

MAINVIEW[®] SRM Allocation EasyPOOL User Guide and Reference

Version 7.2

June 20, 2003



Copyright 2003 BMC Software, Inc., as an unpublished work. All rights reserved.

BMC Software, the BMC Software logos, and all other BMC Software product or service names are registered trademarks or trademarks of BMC Software, Inc. IBM and DB2 are registered trademarks of International Business Machines Corp. All other registered trademarks or trademarks belong to their respective companies.

THE USE AND CONTENTS OF THIS DOCUMENTATION ARE GOVERNED BY THE SOFTWARE LICENSE AGREEMENT ENCLOSED AT THE BACK OF THIS DOCUMENTATION.

Restricted Rights Legend

U.S. GOVERNMENT RESTRICTED RIGHTS. UNPUBLISHED -- RIGHTS RESERVED UNDER THE COPYRIGHT LAWS OF THE UNITED STATES. Use, duplication, or disclosure by the U.S. Government is subject to restrictions set forth in FAR Section 52.227-14 Alt. III (g)(3), FAR Section 52.227-19, DFARS 252.227-7014 (b) or DFARS 227.7202, as amended from time to time. Contractor/Manufacturer is BMC Software, Inc., 2101 CityWest Blvd., Houston, TX 77042-2827, USA. Any contract notices should be sent to this address.

Contacting BMC Software

You can access the BMC Software Web site at <http://www.bmc.com>. From this Web site, you can obtain information about the company, its products, corporate offices, special events, and career opportunities.

United States and Canada

Address BMC Software, Inc.
2101 CityWest Blvd.
Houston TX 77042-2827

Telephone 713 918 8800 or
800 841 2031

Fax 713 918 8000

Outside United States and Canada

Telephone (01) 713 918 8800

Fax (01) 713 918 8000

Customer Support

You can obtain technical support by using the Support page on the BMC Software Web site or by contacting Customer Support by telephone or e-mail. To expedite your inquiry, please see “Before Contacting BMC Software.”

Support Web Site

You can obtain technical support from BMC Software 24 hours a day, 7 days a week at <http://www.bmc.com/support.html>. From this Web site, you can

- read overviews about support services and programs that BMC Software offers
- find the most current information about BMC Software products
- search a database for problems similar to yours and possible solutions
- order or download product documentation
- report a problem or ask a question
- subscribe to receive e-mail notices when new product versions are released
- find worldwide BMC Software support center locations and contact information, including e-mail addresses, fax numbers, and telephone numbers

Support by Telephone or E-mail

In the United States and Canada, if you need technical support and do not have access to the Web, call 800 537 1813. Outside the United States and Canada, please contact your local support center for assistance. To find telephone and e-mail contact information for the BMC Software support center that services your location, refer to the Contact Customer Support section of the Support page on the BMC Software Web site at www.bmc.com/support.html.

Before Contacting BMC Software

Before you contact BMC Software, have the following information available so that Customer Support can begin working on your problem immediately:

- product information
 - product name
 - product version (release number)
 - license number and password (trial or permanent)
- operating system and environment information
 - machine type
 - operating system type, version, and service pack or other maintenance level such as PUT or PTF
 - system hardware configuration
 - serial numbers
 - related software (database, application, and communication) including type, version, and service pack or maintenance level
- sequence of events leading to the problem
- commands and options that you used
- messages received (and the time and date that you received them)
 - product error messages
 - messages from the operating system, such as `file system full`
 - messages from related software

Contents

About This Book	xi
Chapter 1	What is MAINVIEW SRM EasyPOOL?
Chapter 2	How to Use EasyPOOL
Chapter 3	EasyPOOL Functions
Overview	3-2
EMC SymmAPI-MF V510 Support.	3-2
DASDPOOL—Allocate Data Sets to DASD Volume Pools	3-3
Usage Notes	3-17
DSNCHECK—Enforce Data Set Naming Standards	3-30
Usage Notes	3-34
FDRASIST—Pool Data Sets Recalled With FDR/ABR	3-38
Usage Notes	3-42
FORCECAT—Force Catalog Disposition for New Data Sets	3-44
Usage Notes	3-45
MODDELET—Remove Specific Volume for MOD-DELETE Data Set	
Deletion	3-47
Usage Notes	3-48
OPENEMPT—Initialize Empty Data Sets	3-49
Usage Notes	3-50
OPTBLKSZ—Set Optimum Blocksize	3-53
Usage Notes	3-55
SETEXPDT—Set Expiration Date for Data Set.	3-58
Usage Notes	3-61
SMSACSTE—Test ACS Routines	3-64
Usage Notes	3-65
SMSMCREN—Identify Data Set Rename Requiring Management Class	
Change	3-68
Usage Notes	3-69
SMSSELECT—Control Selection of Volumes from SMS Storage Groups	3-72
Usage Notes	3-83
SPACCONV—Change Allocation Units to Blocks	3-90

Usage Notes	3-92
SPACLIMI—Limit the Size of Space Allocation Requests	3-95
Usage Notes	3-97
SPACRLSE—Release Data Set Space at Close	3-100
Usage Notes	3-101
SPACSQTY—Set Primary and Secondary Space for Data Set.	3-104
Usage Notes	3-108
STKSUPP—Split Unit Affinity Specifications Over Devices	3-112
Usage Notes	3-114
SUPJSCAT—Suppress JOBCAT and STEPCAT DD Statements	3-118
Usage Notes	3-119
SUPVOLRF—Suppress Volume References	3-121
Usage Notes	3-122
TAPECOMP—Control Tape Data Set Compaction (IDRC).	3-125
Usage Notes	3-126
TAPEDEFR—Defer Tape Mounts	3-128
Usage Notes	3-129
TAPEPOOL—Assign Tape Data Sets to Pools.	3-130
Usage Notes	3-133
VIOALLOC—Control VIO Allocation	3-136
Usage Notes	3-137
VSAMCNTL—Enforce VSAM Allocation Standards.	3-139
Usage Notes	3-143

Chapter 4

How to Define Critical Data Set Lists

How to Use the CritList Option.	4-3
--------------------------------------	-----

Chapter 5

How to Define DFSMS Subpools

How to Use the SMS Pools Option.	5-4
---------------------------------------	-----

Index

List of Figures

Figure 3-1	MVOLPER View	3-25
Figure 3-2	Possible System Storage Configuration	3-27
Figure 3-3	MVOLPER View	3-86
Figure 5-1	Possible System Storage Configuration for SMS-Managed Subpools	5-3

List of Tables

Table 2-1	Application Groupings of EasyPOOL Functions	2-2
Table 2-2	EasyPOOL Functions and System Parameters Cross-Reference	2-3
Table 3-1	System Parameters—DASDPOOL	3-4
Table 3-2	VSAM Component Separation Conditions	3-18
Table 3-3	System Parameters—DSNCHECK	3-30
Table 3-4	System Parameters—FORCECAT	3-44
Table 3-5	System Parameters—OPTBLKSZ	3-53
Table 3-6	System Parameters—SETEXPDT	3-58
Table 3-7	ACS Variables Displayed by SMSACSTE	3-66
Table 3-8	System Parameters—SMSSELCT	3-73
Table 3-9	Volume Selection Techniques	3-83
Table 3-10	System Parameters—SPACCONV	3-90
Table 3-11	System Parameters—SPACLIMI	3-95
Table 3-12	System Parameters—STKSUPP	3-112
Table 3-13	System parameters—TAPEPOOL	3-130
Table 5-1	SET Statement SMS-Managed Subpool Parameter Quick Reference	5-2
Table 5-2	INC/EXC Statement Pool Parameter Quick Reference	5-2

About This Book

This book contains detailed information about MAINVIEW® Storage Resource Manager Allocation EasyPOOL by BMC Software and is intended for storage administrators.

To use this book, you should be familiar with the following:

- OS/390 operating system
- job control language (JCL)
- Interactive System Productivity Facility (ISPF)

Throughout this book, references to OS/390 support also include support for MVS and z/OS.

How This Book Is Organized

This book is organized as follows. In addition, an index appears at the end of the book.

Chapter/Appendix	Description
Chapter 1, "What is MAINVIEW SRM EasyPOOL?"	provides a brief overview of the EasyPOOL product
Chapter 2, "How to Use EasyPOOL"	lists the functions grouped according to general applications
Chapter 3, "EasyPOOL Functions"	provides detailed information and examples of each EasyPOOL function
Chapter 4, "How to Define Critical Data Set Lists"	provides detailed information about how to define critical data sets
Chapter 5, "How to Define DFSMS Subpools"	provides detailed information about how to define DFSMS subpools

Related Documentation

BMC Software products are supported by several types of documentation:

- online and printed books
- online Help
- release notes and other notices

In addition to this book and the online Help, you can find useful information in the publications listed in the following table. As “Online and Printed Books” on page xii explains, these publications are available on request from BMC Software.

Category	Document	Description
general	<i>MAINVIEW Products General Information</i>	provides an overview of the MAINVIEW environment and the products that it supports
MAINVIEW common documents	<i>OS/390 and z/OS Installer Guide</i> <i>MAINVIEW Installation Requirements Guide</i> <i>MAINVIEW Common Customization Guide</i> <i>Using MAINVIEW</i> <i>MAINVIEW Administration Guide</i> <i>Implementing Security for MAINVIEW</i>	provide instructions for installing, configuring, using, and administering MAINVIEW
MAINVIEW SRM customization documents	<i>MAINVIEW SRM Customization Guide</i>	provides instructions for configuring and customizing MAINVIEW SRM for OS/390 including EasyPOOL
core documents	<i>MAINVIEW SRM User Guide and Reference</i>	provides information common to all MAINVIEW SRM products and high-level navigation
	<i>MAINVIEW SRM Reference Summary</i>	provides a reference of global parameters, filter list and rule list parameters, and functions
messages	<i>MAINVIEW SRM Messages</i>	provides hardcopy of messages that are also available online
supplemental documents	release notes, flashes, technical bulletins	provides additional information about the product

Online and Printed Books

The books that accompany BMC Software products are available in online format and printed format. If you are a Windows or Unix user, you can view online books with Acrobat Reader from Adobe Systems. The reader is provided at no cost, as explained in “To Access Online Books.” You can also obtain additional printed books from BMC Software, as explained in “To Request Additional Printed Books.”

To Access Online Books

Online books are formatted as Portable Document Format (PDF) files. You can view them, print them, or copy them to your computer by using Acrobat Reader 3.0 or later. You can access online books from the documentation compact disc (CD) that accompanies your product or from the World Wide Web.

In some cases, installation of Acrobat Reader and downloading the online books is an optional part of the product-installation process. For information about downloading the free reader from the Web, go to the Adobe Systems site at <http://www.adobe.com>.

To view any online book that BMC Software offers, visit the support page of the BMC Software Web site at <http://www.bmc.com/support.html>. Select a product to access the related documentation.

To Request Additional Printed Books

BMC Software provides printed books with your product order. To request additional books, go to <http://www.bmc.com/support.html>.

Release Notes and Other Notices

Printed release notes accompany each BMC Software product. Release notes provide current information such as

- updates to the installation instructions
- last-minute product information

In addition, BMC Software sometimes provides updated product information between releases (in the form of a flash or a technical bulletin, for example). The latest versions of the release notes and other notices are available on the Web at <http://www.bmc.com/support.html>.

Conventions

This section provides examples of the conventions used in this book and explains how to read ISPF panel-flow diagrams and syntax statements.

General Conventions

This book uses the following general conventions:

Item	Example
information that you are instructed to type	Type SEARCH DB in the designated field.
specific (standard) keyboard key names	Press Enter .
field names, text on a panel	Type the appropriate entry in the Command field.
directories, file names, Web addresses	The BMC Software home page is at www.bmc.com .
nonspecific key names, option names	Use the HELP function key. KEEPDICTIONARY option
MVS calls, commands, control statements, keywords, parameters, reserved words	Use the SEARCH command to find a particular object. The product generates the SQL TABLE statement next.
code examples, syntax statements, system messages, screen text	//STEPLIB DD The table <i>table_name</i> is not available.
emphasized words, new terms, variables	The instructions that you give to the software are called <i>commands</i> . In this message, the variable <i>file_name</i> represents the file that caused the error.
single-step procedures	»» To enable incremental backups, type y and press Enter at the next prompt.

This book uses the following types of special text:

Note: Notes contain important information that you should consider.

Warning! Warnings alert you to situations that could cause problems, such as loss of data, if you do not follow instructions carefully.

Tip: Tips contain useful information that may improve product performance or that may make procedures easier to follow.

Syntax Statements

Syntax statements appear in Courier. The following example shows a sample syntax statement:

```
COMMAND KEYWORD1 [KEYWORD2|KEYWORD3] KEYWORD4={YES|NO}  
    file_name...
```

The following table explains conventions for syntax statements and provides examples:

Item	Example
Items in italic type represent variables that you must replace with a name or value. Use an underscore for variables with more than one word.	<code>dtbackup <i>control_directory</i></code>
Brackets indicate a group of options. You can choose at least one of the items in the group, but none of them is required. Do not type the brackets when you enter the option. A comma means that you can choose one or more of the listed options. You must use a comma to separate the options if you choose more than one option.	<code>[<i>table_name, column_name, field</i>]</code>
Braces enclose a list of required items. You must enter at least one of the items. Do not type the braces when you enter the item.	<code>{<i>DBD_name table_name</i>}</code>
A vertical bar means that you can choose only one of the listed items. In the example, you would choose either <i>commit</i> or <i>cancel</i> .	<code>{commit cancel}</code>
An ellipsis indicates that you can repeat the previous item or items as many times as necessary.	<code><i>column_name . . .</i></code>

Chapter 1 What is MAINVIEW SRM EasyPOOL?

MAINVIEW SRM EasyPOOL is one of the products that make up the MAINVIEW SRM Allocation storage management product group, which also includes MAINVIEW SRM StopX37/II and MAINVIEW SRM EasySMS.

The Allocation component provides

- enhancements to OS/390 space management, reducing the incidence of space-related processing problems
- control over data set allocation and enforcement of allocation and naming standards
- tools that aid in the conversion to DFSMS and provide enhancements to the DFSMS environment after implementation

Allocation functions operate at the operating system level to intercept abend conditions or standards violations, thus providing services without any JCL changes.

The following are some of the benefits and uses of EasyPOOL:

- allocates new data sets to pools instead of individual volumes
- allocates temporary data sets to VIO
- limits allocated space
- enforces standards by controlling:
 - data set naming
 - cataloging
 - data set expiration date usage
 - space allocation units (blocks versus tracks/cylinders)
 - release of unused space
 - tape data set compaction

-
- optimizes block sizes
 - suppresses JOBCAT and STEPCAT DD statements
 - organizes tape devices into pools
 - redirects tape data sets to DASD and vice versa
 - performs affinity separation of tape volumes in/out of silos
 - isolates critical data sets

In addition, volume selection can be based on user-defined criteria (maximum space, EMC volume, performance, and so on). Data sets can be pooled based on historical performance data collected for up to two weeks.

Chapter 2 How to Use EasyPOOL

EasyPOOL consists of a set of functions that help you

- pool data sets
- enforce data center standards
- enhance tape functionality
- enhance DFSMS functionality
- enhance data set allocation efficiency
- enhance volume selection
- isolate critical data sets
- run DASD Performance Optimization (DPO)
- run Historical DASD Performance Optimization

Using EasyPOOL involves activating and controlling EasyPOOL functions. Refer to the *MAINVIEW SRM User Guide and Reference* for details.

Note: Pools must be defined prior to setting up EasyPOOL functions. Refer to the *MAINVIEW SRM User Guide and Reference* for information about defining pools.

The following table lists the functions grouped according to general applications. See “EasyPOOL Functions and System Parameters Cross-Reference” on page 2-3 for a matrix of EasyPOOL functions and the system parameters that support them.

Table 2-1 Application Groupings of EasyPOOL Functions

If you need to...	Consider using...
pool data sets	"DASDPOOL—Allocate Data Sets to DASD Volume Pools" on page 3-3 "FDRASIST—Pool Data Sets Recalled With FDR/ABR" on page 3-38 "MODDELETE—Remove Specific Volume for MOD-DELETE Data Set Deletion" on page 3-47 "SUPVOLRF—Suppress Volume References" on page 3-121 "VIOALLOC—Control VIO Allocation" on page 3-136
enforce data center standards	"DSNCHECK—Enforce Data Set Naming Standards" on page 3-30 "FORCECAT—Force Catalog Disposition for New Data Sets" on page 3-44 "SUPJSCAT—Suppress JOBCAT and STEPCAT DD Statements" on page 3-118 "VSAMCNTL—Enforce VSAM Allocation Standards" on page 3-139
enhance tape functionality	"STKSUPP—Split Unit Affinity Specifications Over Devices" on page 3-112 "TAPECOMP—Control Tape Data Set Compaction (IDRC)" on page 3-125 "TAPEDEFR—Defer Tape Mounts" on page 3-128 "TAPEPOOL—Assign Tape Data Sets to Pools" on page 3-130
enhance DFSMS functionality	"SMSACSTE—Test ACS Routines" on page 3-64 "SMSMCREN—Identify Data Set Rename Requiring Management Class Change" on page 3-68 "SMSSELECT—Control Selection of Volumes from SMS Storage Groups" on page 3-72
enhance data set allocation efficiency	"OPENEMPT—Initialize Empty Data Sets" on page 3-49 "OPTBLKSZ—Set Optimum Blocksize" on page 3-53 "OPTBLKSZ—Set Optimum Blocksize" on page 3-53 "SETEXPDT—Set Expiration Date for Data Set" on page 3-58 "SPACCONV—Change Allocation Units to Blocks" on page 3-90 "SPACLIMI—Limit the Size of Space Allocation Requests" on page 3-95 "SPACRLSE—Release Data Set Space at Close" on page 3-100 "SPACSQTY—Set Primary and Secondary Space for Data Set" on page 3-104
enhance volume selection	"DASDPOOL—Allocate Data Sets to DASD Volume Pools" on page 3-3 "SMSSELECT—Control Selection of Volumes from SMS Storage Groups" on page 3-72
isolate critical data sets	"DASDPOOL—Allocate Data Sets to DASD Volume Pools" on page 3-3 "SMSSELECT—Control Selection of Volumes from SMS Storage Groups" on page 3-72
run DASD Performance Optimization (DPO)	"DASDPOOL—Allocate Data Sets to DASD Volume Pools" on page 3-3 "SMSSELECT—Control Selection of Volumes from SMS Storage Groups" on page 3-72
run Historical DASD Performance Optimization (HISTDPO)	"DASDPOOL—Allocate Data Sets to DASD Volume Pools" on page 3-3 "SMSSELECT—Control Selection of Volumes from SMS Storage Groups" on page 3-72

Some functions within EasyPOOL are affected by specific system parameters. The following table lists system parameters and the EasyPOOL functions affected by them.

Table 2-2 EasyPOOL Functions and System Parameters Cross-Reference

System Parameter	DASDPOOL	DSNCHECK	FDRASIST	FORCECAT	MODDELET	OPENEMPT	OPTBLKSZ	SETEXPDT	SMACSTE	SMSMCREN	SMSSELC	SPACCONV	SPACLIMI	SPACRLSE	SPACQTY	STKSUPP	SUPJSCAT	SUPVOLRF	TAPECOMP	TAPEDEFR	TAPEPOOL	VIOALLOC	VSAMCNTL
BLKINPUT							X																
BLKOLDSR							X																
CHECK	X																						
CRITLIST	X										X												
DASDGENR	X	X		X				X															
DMYUNIT	X																				X		
DP_RENAME	X																						
HISTDAYS	X										X												
JCLEXT	X																						
JCLUREQ	X																						
PROCOLD	X																						
REJECT	X	X											X								X		
SMS_ALLOC											X												
SMS_EXTEND											X												
STKSCR																	X						
TAPEGENR		X		X				X													X		
TRKCYL							X					X											
TRKLEN							X					X											

Chapter 3 EasyPOOL Functions

The following topics are discussed:

Overview	3-2
DASDPOOL—Allocate Data Sets to DASD Volume Pools	3-3
DSNCHECK—Enforce Data Set Naming Standards	3-30
FDRASIST—Pool Data Sets Recalled With FDR/ABR	3-38
FORCECAT—Force Catalog Disposition for New Data Sets	3-44
MODDELET—Remove Specific Volume for MOD-DELETE Data Set Deletion	3-47
OPENEMPT—Initialize Empty Data Sets	3-49
OPTBLKSZ—Set Optimum Blocksize	3-53
SETEXPDT—Set Expiration Date for Data Set	3-58
SMSACSTE—Test ACS Routines	3-64
SMSMCREN—Identify Data Set Rename Requiring Management Class Change	3-68
SMSSELECT—Control Selection of Volumes from SMS Storage Groups	3-72
SPACCONV—Change Allocation Units to Blocks	3-90
SPACLIMI—Limit the Size of Space Allocation Requests	3-95
SPACRLSE—Release Data Set Space at Close	3-100
SPACSQTY—Set Primary and Secondary Space for Data Set	3-104
STKSUPP—Split Unit Affinity Specifications Over Devices	3-112
SUPJSCAT—Suppress JOBCAT and STEPCAT DD Statements	3-118
SUPVOLRF—Suppress Volume References	3-121
TAPECOMP—Control Tape Data Set Compaction (IDRC)	3-125
TAPEDEFR—Defer Tape Mounts	3-128
TAPEPOOL—Assign Tape Data Sets to Pools	3-130
VIOALLOC—Control VIO Allocation	3-136
VSAMCNTL—Enforce VSAM Allocation Standards	3-139

Overview

MAINVIEW SRM storage management services are divided into functions. SMFUNC xx activates MAINVIEW SRM functions and controls message and tracing activity. Functions provide all the runtime services of MAINVIEW SRM. SMFUNC xx points to members SMFLST xx and SMRLST xx that select resources and control the operation of the functions.

Functions are defined in SET statements. You can change parameters

- by editing the member directly
- through the Functions option on the Parmlib menu of the EZSRM Menu (see the *MAINVIEW SRM User Guide and Reference*).

EasyPOOL functions do not operate for DFDSS jobs.

EMC SymmAPI-MF V510 Support

MAINVIEW SRM supports EMC ResourcePak Base product and EMC micro code levels 5x68 and up.

MAINVIEW SRM uses the EMC SymmAPI to obtain extended EMC information, which includes the mapping of a logical user volume to an internal physical disk. The SymmAPI running on OS/390 communicates with the hardware micro code installed within the Symmetrix device. Both the hardware micro code and the software SymmAPI can be at various versions/levels. A given API version can only communicate with certain micro code levels. Generally speaking, a new version of the API will handle whatever the current micro code level is, and all micro code levels prior to it. From the micro code level point of view, for each micro code level there is a minimum API level requirement. The SymmAPI V410 software distributed within MAINVIEW SRM supports micro code level 5x67 and below.

Warning! The EMC SymmAPI functions are not available when MAINVIEW SRM is running on a VM guest machine. When run on a VM guest, MAINVIEW SRM issues a message to the SVOS joblog during product start up indicating extended EMC data will be unavailable.

DASDPOOL—Allocate Data Sets to DASD Volume Pools

The DASDPOOL function is used to assign data sets to pools of DASD volumes. Pools allow multiple volumes to be managed as single entities. Pools allow data sets to be grouped and managed by their characteristics. Pool storage groups are a key component of DFSMS.

Considerations in DASD pooling include:

- use of storage versus private volumes
- volume allocation for non-specific requests
- acceptance/rejection of requests for allocation on specific volumes
- grouping of data sets by usage, by size, or by physical characteristics

Assignment of data sets to pools should be centrally controllable by the data center, without requiring JCL changes or any type of source changes. The storage administrator should be able to easily manage the storage of the entire organization.

DASDPOOL allows you to set standards for data sets' volume residency. A collection of data sets, selected by filter and rule list parameters from a variety of job and data set properties, can be directed to reside on the volumes in a single pool or in multiple pools for more flexibility in space utilization and management.

Critical Data Set Isolation

EasyPOOL allows you to identify a list of critical data sets in your environment that should never be placed on the same volume as any other data set in the list. EasyPOOL may also use data gathered by the Space Collector about the EMC Symmetrix environments to ensure that critical data sets are not only on separate logical volumes but are placed on separate physical volumes to reduce data set contention.

Note: The Space Collector must be installed and running to access EMC Symmetrix data.

It is important to consider, however, that CRITDSN is resource intensive and should be used only for a *small* list of critical data sets. It should not be used without considering the system impact. See Chapter 4, “How to Define Critical Data Set Lists,” for information on defining the SMCRITxx definition parameter.

Note: The DASDPOOL function must be specified in the SMFUNCxx member to be available and must be set active to provide services.

System Parameters

The system parameters that affect DASDPOOL are described in Table 3-1.

Table 3-1 System Parameters—DASDPOOL (Part 1 of 2)

Parameter	Purpose
CHECK	Specifies candidate volume replacement during allocation for multi-volume data sets. If multiple volumes are requested for a data set, VSAM or non-VSAM, the CHECK parameter controls whether the first volume or all volumes are immediately assigned from the assigned pool.
CRITLIST	Specifies the suffix of an SMCRTxx parameter member.
DASDGENR	Specifies the only generic DASD names that are to be processed for specific functions if no space requirements are specified. Standard pooling functionality allows the JCL UNIT parameter to specify a pool name. If this is not desired, DASDGENR can be used to intercept data set allocations without space information. Such allocations will only be considered for subsequent processing if the unit is found in an internal table (3380, 3350, and so on) or if the unit is found in the DASDGENR list. DASDGENR affects functions DASDPOOL, DSNCHECK, SETEXPDT, and FORCECAT. If no DASDGENR table is coded, all DASD allocations will be processed.
DMYUNIT	Defines the conversion of a nonexistent UNIT parameter to a valid UNIT parameter only if JCLEXT=YES.
DP_RENAME	Indicates whether DASDPOOL is to be processed on renamed data sets.
HISTDAYS	Specifies the number of days of data to retrieve from the Performance Collector and temporarily store in a data space. Values are 0-14 (7 or 14 recommended), which indicate the number of days of data to be retrieved. The default value is 0, which indicates that the data space is created, but no historical performance data is gathered and no HISTDPO pooling can be performed. If this value is changed, the Performance Collector (SVSGP) must be stopped and restarted. For a valid HISTDPO test, do not specify a number of HISTDAYS higher than the total number of days collected in the Performance Collector.
JCLEXT	Specifies if EasyPOOL will obtain volume and unit information after the operating system accesses the catalog. This is useful when PROCOLD=YES and/or DISP=MOD is specified.
JCLUREQ	When PROCOLD=YES is specified and EasyPOOL is analyzing a DD statement for an existing data set, the JCLUREQ parameter determines if UNIT information is required. If JCLUREQ=YES is specified, MAINVIEW SRM intercepts the DD statement only when the JCL specifies UNIT parameter. This allows the data center to correct JCL that uses an invalid unit parameter for existing data sets without analyzing DD statements that are correctly specified. EasyPOOL examines every DD statement associated with an existing data set if you specify JCLUREQ=NO. Note: JCLUREQ=NO needs to be specified when processing UNIT=AFF groups since the unit field in the SIOT consists of blanks.

Table 3-1 System Parameters—DASDPOOL (Part 2 of 2)

Parameter	Purpose
PROCOLD	Specifies if EasyPOOL will intercept DD statements that specify OLD allocations. EasyPOOL always intercepts NEW and MOD allocations. EasyPOOL also intercepts DD statements that specify the UNIT parameter when PROCOLD=YES is specified, which is useful for correcting questionable JCL. For example, assume DASDPOOL pools a data set to a TEST volume. If a later step wants to use the data set and specifies UNIT=PROD, the job receives a JCL error. You can correct this situation by specifying PROCOLD=YES.
REJECT	Controls termination of processing at the first or last data set rejected by REJECT=YES in DASDPOOL or DSNCHECK. If termination is to take place on the first rejected data set, code REJECT=FIRST. If all data sets are to be processed before control is returned to OS/390 allocation, code REJECT=LAST.

Rule List Parameter Quick Reference

Parameter	Description
CANDIDATE=YES/NO	Candidate volumes accepted (VSAM data sets)
CRITEMC=YES/NO	Specifies whether the volume meeting CRITDSN criteria includes EMC physical volumes
CRITBIAS=nnnn	Defines the number of data sets that can reside on the volume before the current allocation
CRITFAIL=YES/NO	Defines allocation if a volume meeting the criteria cannot be found
CRITLIST=xxxxxxxx	Name of the table containing allocation volumes
DPORDEF=nnn	Default response time for device selection
DPORMAX=nn	Maximum response time target for device selection
DPORMIN=nn	Minimum response time target for device selection
DPORSEP=nn	Separation factor for device selection
DPOWIND=nnnn	Interval for device selection based on performance
EVENTID=xxxxx	Identifies a user event defined in an SMEVNTxx member
HDPORDEF=nnn	Default response time for device selection
HDPORMAX=nn	Maximum response time target for device selection
HDPORMIN=nn	Minimum response time target for device selection
HDPORSEP=nn	Separation factor for device selection
HDPOSTIM=nnnn	Starting time each day for a range of snapshots to be analyzed
HDPOETIM=nnnn	Ending time each day for a range of snapshots to be analyzed
HDPODAYS=(MO,TU,WE,TH,FR,SA,SU)	Day(s) of the week used in determining performance statistics for pooling
NVOLMAX=YES/NO	Limits volumes to the number in the assigned pool
NVOL=nn	Number of volumes and units that can be allocated to a data set
NVOLINDX=nn	Number of volumes that can be allocated to a VSAM index component
POOL=(xxxxxxxx,...)	1-15 pool names (from SMPPOOLxx) (1-8 characters)

Parameter	Description
RAIDDEVTYPE=xxxxxxxxx	Allows the user to require a specific RAID device type for an allocation (EMC/RDFEMC/MIRROREMC/PARITYEMC) (requires the MAINVIEW SRM Reporting component) For RAIDDEVTYPE to be honored, a VOLSEL routine <i>must be</i> assigned.
REJECT=YES/NO	Rejects recommended volume assignment
SEP=YES/NO/ASIS	Specifies volume separation of data/index VSAM components
SUPVOL=YES/NO	Suppresses request for allocation to a specific volume
UNIT=xxxxxxxxx	Unit name (esoteric or generic) 1-8 characters
USEVOL=STOR/PRIV/ALL	Directs volume allocation to a specific type (STOR, PRIV, all)
VFORCE=YES/NO	Adds standard component suffixes (DATA, INDEX) to non-JCL defined VSAM file names
VOLSEL=xxxxxxxxx	Volume selection criteria (BESTFIT, CRITDSN, DPO, HISTDPO, MAXSPACE, PERCENT)

Rule List Parameter Explanations

CANDIDATE=

Purpose: Specifies whether the IDCAMS DEFINE VOL=() list for a VSAM data set is used as an indicator of secondary volume allocation. If CANDIDATE=YES, the *number* of DEFINEd volumes is used to select that many candidate volumes from the MAINVIEW SRM pool. If CANDIDATE=NO, no secondary volumes are assigned or available through normal OS/390 processing (but SPACVOLA can be used to assign secondary volumes from a MAINVIEW SRM pool). Note that with specification of the CANDIDATE parameter, the candidate volumes are assigned from the pool, not from those volumes specified in the IDCAMS DEFINE VOL list. CANDIDATE overrides NVOL and NVOLINDX. The CANDIDATE parameter does not apply to JCL-defined VSAM data sets.

Syntax: CANDIDATE=YES/NO

Default: CANDIDATE=NO

CRITBIAS=

Purpose: Defines the number of data sets from the assigned CRITLIST that can reside on the volume before the current allocation.

Syntax: CRITBIAS=xxxx

where xxxx is a number 0–9999.

Default: None

Note: This parameter works only in conjunction with VOLSEL=CRITDSN.

CRITEMC=

Purpose: Specifies whether the volume meeting CRITDSN criteria includes EMC physical volumes. If yes, data sets specified with CRITDSN should not reside on the same EMC Physical volume nor the same MVS Logical volume. If no, data sets specified with CRITDSN may reside on the same EMC physical volume. Default is no.

Syntax: CRITEMC=*YES/NO*

Default: CRITEMC=NO

Note: This parameter works only in conjunction with VOLSEL=CRITDSN.

CRITFAIL=

Purpose: Defines the allocation process if a volume meeting the CRITDSN criteria cannot be found. If Yes, allocation fails; if No, allocation is made to the best available volume. Like USELIM, if an acceptable volume cannot be found in the first pool, the best volume from the first pool is saved, and the system tries the next pool(s). If no acceptable volume is found, CRITFAIL is processed on the saved volume.

Syntax: CRITFAIL=*YES/NO*

Default: CRITFAIL=NO

Note: This parameter works only in conjunction with VOLSEL=CRITDSN.

CRITLIST=

Purpose: The table that contains a list of critical data sets.

Syntax: CRITLIST=*xxxxxxxx*

Default: None

Note: This parameter works only in conjunction with VOLSEL=CRITDSN.

DPORDEF=

Purpose: Specifies the default response time for device selection based on DASD performance. The default value is used when response time information has not been accumulated for a device, such as when it has just been brought online.

DPORDEF is useful only when specified in conjunction with DPORMIN/MAX.

Syntax: DPORDEF=*nnn*

where *nnn* specifies the default response time in milliseconds that will be substituted for unavailable information for a specific device.

Default: DPORDEF=50

DPORMAX=

Purpose: Specifies the maximum response time objective for device selection based on DASD performance.

DPORMAX is normally specified in conjunction with DPORMIN.

Syntax: DPORMAX=*nn*

where *nn* specifies the maximum response time in milliseconds that will be considered for device selection based on performance.

Default: None

DPORMIN=

Purpose: Specifies the minimum response time objective for device selection based on DASD performance.

DPORMIN is normally specified in conjunction with DPORMAX.

Syntax: DPORMIN=*nn*

where *nn* specifies the minimum response time in milliseconds that will be considered for device selection based on performance.

Default: None

DPORSEP=

Purpose: Specifies the response time value that is used to force selection of different volumes for data sets in the same jobstep. When multiple data sets are allocated in a single jobstep and device selection based on DASD performance is specified for some or all, this separation factor is added to the current response time for previously used volumes in order to increase the likelihood that new allocations will go to a different volume.

Syntax: DPORSEP=*nn*

where *nn* specifies the response time in milliseconds that will be used to enhance data set separation across volumes.

Default: DPORSEP=10

DPOWIND=

Purpose: Specifies the window (length of the performance interval) that is used to analyze the response characteristics of DASD devices for selection based on performance.

Syntax: DPOWIND=*nnnn*

where *nnnn* specifies the number of seconds over which the performance analysis will be made.

Default: DPOWIND=1800 (this is the maximum data retained in the performance tables)

EVENTID=

Purpose: Specifies the identifier assigned to a user event in SMEVNT_{xx}. This parameter will cause an event to be generated from this function.

Note: If EVENTID= is used on an FLST SET statement with MODE=INACT, the event will still be issued.

Syntax: EVENTID=*xxxxx*

where *xxxxx* is the 5-character string specified on the EVNTID parameter in SMEVNT_{xx}.

Default: None

HDPORDEF=

Purpose: Specifies the default response time for device selection based on DASD performance. The default value is used when response time information has not been accumulated for a device, such as when it has just been brought online.

HDPORDEF is useful only when specified in conjunction with HDPORMIN/MAX. This parameter is available only when VOLSEL=HISTDPO.

Syntax: HDPORDEF=*nnn*

where *nn* specifies the default response time in milliseconds that will be substituted for unavailable information for a specific device.

Default: HDPORDEF=50

HDPORMAX=

Purpose: Specifies the maximum response time objective for device selection based on DASD performance.

HDPORMAX is normally specified in conjunction with HDPORMIN. This parameter is available only when VOLSEL=HISTDPO.

Syntax: HDPORMAX=*nn*

where *nn* specifies the maximum response time in milliseconds that will be considered for device selection based on performance.

Default: None

HDPORMIN=

Purpose: Specifies the minimum response time objective for device selection based on DASD performance.

HDPORMIN is normally specified in conjunction with HDPORMAX. This parameter is available only when VOLSEL=HISTDPO.

Syntax: HDPORMIN=*nn*

where *nn* specifies the minimum response time in milliseconds that will be considered for device selection based on performance.

Default: None

HDPORSEP=

Purpose: Specifies the response time value that is used to force selection of different volumes for data sets in the same jobstep. When multiple data sets are allocated in a single jobstep and device selection based on DASD performance is specified for some or all, this separation factor is added to the current response time for previously used volumes in order to increase the likelihood that new allocations will go to a different volume. This parameter is available only when VOLSEL=HISTDPO.

Syntax: HDPORSEP=*nn*

where *nn* specifies the response time in milliseconds that will be used to enhance data set separation across volumes.

Default: HDPORSEP=10

HDPOSTIM=

Purpose: Specifies the starting time (each day) for a range of snapshots to be analyzed. This allows analysis of data from the same time range, or shift, each day. This parameter is available only when VOLSEL=HISTDPO.

Syntax: HDPOSTIM=*nnnn*

where *nnnn* specifies a time in 24-hour format.

Default: HDPOSTIM=0000

HDPOETIM=

Purpose: Specifies the ending time (each day) for a range of snapshots to be analyzed. This allows analysis of data from the same time range, or shift, each day. This parameter is available only when VOLSEL=HISTDPO.

Syntax: HDPOETIM=*nnnn*

where *nnnn* specifies a time in 24-hour format.

Default: HDPOETIM=2359

HDPODAYS=

Purpose: Specifies the day(s) of the week to be used in determining performance statistics for pooling. This parameter is available only when VOLSEL=HISTDPO.

Syntax: HDPODAYS=(*MO, TU, WE, TH, FR, SA, SU*)

Default: HDPODAYS=(*MO, TU, WE, TH, FR, SA, SU*)

NVOL=

Purpose: Specifies the number of volumes and units that can be allocated to a data set.

Syntax: NVOL=*nn*

where *nn* is a number in the range of 1–59 for non-VSAM data sets and 1–20 for VSAM data components.

Default: None

NVOLINDEX=

Purpose: Specifies the number of volumes that can be allocated to a VSAM index component. Note that CANDIDATE overrides NVOLINDEX for non-JCL defined VSAM data sets.

Syntax: NVOLINDEX=*nn*

where *nn* is a number in the range of 1–20.

Default: None

NVOLMAX=

Purpose: Specifies whether the number of volumes that can be assigned to a data set is limited to the number of volumes in the pool that the data set resides in. Note that volume counts in alternate pools are not included in NVOLMAX.

Syntax: NVOLMAX=*YES/NO*

Default: NVOLMAX=NO

POOL=

Purpose: Specifies the name of the pool to receive the data set. If multiple pools are specified, the first pool with sufficient space for the data set is assigned. (The first pool specified is the primary pool; all others are alternate pools.) Note that a pool specification is not required in certain circumstances, such as when rejecting an allocation (REJECT=YES) or allowing a specific volume request (SUPVOL=NO).

Syntax: POOL=xxxxxxx or POOL=(xxxxxxx,xxxxxxx,...)

where xxxxxxx is a pool name defined in SMPOOLxx. If the poolname is not defined, the refresh or start-up will fail. Up to 15 pools can be assigned within parentheses.

Default: None

RAIDDEVTYPE=

Purpose: Allows the user to require a specific RAID device type for an allocation. If a particular hardware type is chosen, the candidate list of volumes will be limited to devices that have the requested attribute flag enabled. This parameter requires the MAINVIEW SRM Reporting component to be installed and running.

Note: For RAIDDEVTYPE to be honored, a VOLSEL routine *must be* assigned.

Syntax: RAIDDEVTYPE=xxxxxxxx

where xxxxxxx is one of the following:

EMC	The device must be an EMC RAID device.
RDFEMC	The device must be EMC and have the RDF flag enabled.
MIRROREMC	The device must be EMC and have the MIRROR flag enabled.
PARITYEMC	The device must be EMC and have a parity flag enabled.

Default: None

REJECT=

Purpose: Specifies that the recommended device assignment (from OS/390 or DFSMS) is to be rejected. No DASDPOOL processing is performed; the request is terminated immediately, and message SVM3311I is issued. Note that this causes the allocation (and the job) to fail.

Syntax: REJECT=*YES/NO*

Default: REJECT=NO

SEP=

Purpose: Specifies whether the data and index components of a non-JCL defined VSAM key-sequenced data set are allocated to separate volumes in a pool.

CANDIDATE=YES must also be specified for VSAM component separation.

Syntax: SEP=*YES/NO/ASIS*

Default: None

SUPVOL=

Purpose: Specifies whether an allocation request for a specific volume is suppressed. If SUPVOL=YES, the request for a specific volume is ignored, and the data set is assigned to a pool. If SUPVOL=NO, the specific volume request is honored and the data set is not pooled. This parameter applies to both VSAM and non-VSAM data sets.

Syntax: SUPVOL=*YES/NO*

Default: SUPVOL=YES

UNIT=

Purpose: Specifies a unit name (esoteric or generic) to be used by the data set during allocation.

Syntax: UNIT=xxxxxxx

where xxxxxxx is a valid unit name (esoteric or generic) 1–8 characters long.

Default: None

USEVOL=

Purpose: Specifies the type of volume (storage, private, or either) that satisfies non-specific allocation requests. **USEVOL=STOR** directs the data set allocation to a storage-mounted volume; **USEVOL=PRIV** directs the data set allocation to a private volume. (There is currently no specification for **PUBLIC**-only volumes.) **USEVOL=ALL** directs the allocation to the first available volume, regardless of use attribute.

Syntax: **USEVOL=STOR/PRIV/ALL**

STOR	storage mounted volume
PRIV	privately mounted volume
ALL	volume of any mount type

Default: **USEVOL=ALL**

VFORCE=

Purpose: Specifies naming conventions for non-JCL defined VSAM components by adding standard component suffixes (**DATA**, **INDEX**) to VSAM data set cluster names.

Syntax: **VFORCE=YES/NO**

YES	overrides unspecified or invalid VSAM component names by appending .DATA and .INDEX qualifiers to the cluster name
NO	does not modify VSAM component names

Default: **VFORCE=NO**

VOLSEL=

Purpose: Specifies that volume selection within a pool should be based on current volume space availability.

Syntax: *VOLSEL=BESTFIT/CRITDSN/DPO/HISTDPO/MAXSPACE/PERCENT*

Specifies that the volume with the smallest contiguous extent that satisfies the primary allocation should be selected.

BESTFIT specifies that the volume with the smallest contiguous extent that satisfies the primary allocation should be selected

CRITDSN controls allocations by specifying data sets that should not reside on the same volume

Warning! CRITDSN is resource intensive and should be used only for a small list of critical data sets. It should not be used without considering the system impact.

DPO specifies volume selection based on performance statistics accumulated by MAINVIEW SRM

HISTDPO specifies volume selection based on historical volume performance

Note: HISTDPO requires more resources than normal allocations.

MAXSPACE specifies that the volume with the largest single contiguous extent should be selected

PERCENT specifies that the volume with the largest amount of free space should be selected

Default: None

Parameters Not Supported

The following filter list parameters are not supported for DASDPOOL:

- GDGVER
- VALUE

Usage Notes

DASDPOOL provides services for non SMS-managed data sets with a status disposition of NEW or MOD. (The function SMSSELECT performs this service for DFSMS-managed data sets.)

Data sets can be assigned to a single pool, or to a collection of up to 15 pools. When multiple pools are specified, DASDPOOL checks each pool in the order specified to locate sufficient space for the data set. All pools specified must be defined in SMPOOLxx or the refresh or start-up will fail.

DASDPOOL allows pool names to be used as OS/390 esoteric device names, without definition as such in OS/390. Data sets assigned to the DASDPOOL pool name specified in the JCL UNIT parameter will be assigned to a volume in that pool. For example, if pool MAXSIZE is defined in SMPOOLxx, then:

```
//NEWTRANS DD DSN=data-set-name ,  
//DISP=(NEW,CATLG,DELETE) ,  
//UNIT=MAXSIZE ,  
//SPACE=(. . . .
```

would place the NEWTRANS data set on a volume in pool MAXSIZE (MAXSIZE must be a DASD pool). DASDPOOL FLST/RLST rules may then be used to change the pool or any other SET parameter.

DASDPOOL assignments override any JCL-specified UNIT and VOL=SER parameters.

DASDPOOL does not work with allocations made through SVC 32.

VSAM Component Separation

The data and index components of IDCAMS-defined key-sequenced VSAM data sets can be assigned to separate volumes by DASDPOOL through the use of the SEP rule list parameter. Both CANDIDATE=YES and SEP=YES or ASIS must be specified on the rule. The SEP parameter does not apply to JCL-defined VSAM data sets.

SEP=YES causes the data and index components to be assigned to separate volumes in the pool. If the number of volumes defined in the pool is insufficient, alternate pools are searched to find enough volumes to satisfy the request. If no alternate pools are defined, or none of the alternate pools has enough volumes, the allocation fails on the index component.

SEP=ASIS causes the data and index components to be assigned to separate volumes if the original definition of the VSAM cluster specifies different volumes for the two components. The usage of alternate pools is the same as for SEP=YES.

The exact effect of VSAM component separation depends on the way the VSAM data set is defined. Table 3-2 summarizes how the VSAM components are separated under various conditions.

Table 3-2 VSAM Component Separation Conditions

Option	Single Volume Definition	Multi-Volume Definition at the Cluster Level	Multi-Volume Definition at the Component Level
SEP=YES	Data/index separated within pool(s)—if pool has insufficient volumes, alternate pool used—if no alternate pool, index allocation fails	Multiple volumes assigned for both components—number same as in original cluster definition—(SEP has the effect of doubling volume requirements)—if pool has insufficient volumes, alternate pool used—if no alternate pool, index allocation fails	Number of volumes assigned matches that defined in the component definitions—if pool has insufficient volumes, alternate pool used—if no alternate pool, index allocation fails
SEP=NO	No separation processing performed—data/index are assigned to the same volume	No separation processing performed—data/index are assigned to the same volume	No separation processing performed—data/index are assigned to the same volume
SEP=ASIS	If a single volume is defined at cluster level, or the same single volume is defined at the component levels, no separation processing performed—if different single volumes are specified at the component level, processing is same as the Multi-Volume at Component Level column	No separation processing performed—data/index are assigned to the same volume	If the volume lists in both components are identical, processing is as if SEP=NO were specified—if the volume lists differ, processing is as if SEP=YES were specified

The use of alternate pools for VSAM component separation is highly recommended, particularly for multi-volume allocations.

VSAM component separation is not supported for JCL-defined VSAM data sets.

CANDIDATE parameter

The CANDIDATE parameter applies only to non-JCL defined VSAM data sets, for allocation of secondary volumes. If CANDIDATE=YES, DASDPOOL will allocate as many volumes from the selected pool as are in the IDCAMS DEFINE VOL=() list; the first volume selected from the pool will be allocated as the primary, while the other volumes from the pool will be candidate volumes. If CANDIDATE=NO is specified, the IDCAMS DEFINE VOL specification will be ignored, and there will be no secondary volumes available for normal OS/390 secondary volume allocation. (However, the MAINVIEW SRM function SPACVOLA can be used to allocate secondary volumes from the desired pool.) Note that with specification of the CANDIDATE parameter, the volume or volumes are assigned from the pool, not from those volumes specified in the IDCAMS DEFINE VOL list. Note also that CANDIDATE overrides NVOL. The CANDIDATE parameter does not apply to JCL-defined VSAM data sets; the number of volumes can be controlled using the NVOL and NVOLINDX parameters.

SUPVOL parameter

The SUPVOL parameter allows requests for specific volumes to be accepted or rejected. This enables the data center to direct data sets to specific volumes in certain situations, while disallowing it in general. When SUPVOL=NO is specified, a request for a specific volume (from JCL, IDCAMS DEFINE, and so on) is not suppressed; the specific volume requested will be assigned, and the data set will not be pooled. The rule may omit the pool name (SET SUPVOL=NO); however, if a pool name is specified (SET POOL=name SUPVOL=NO), the pool is ignored. When invoked, SUPVOL=NO processing will generate message number SMS3320I.

USEVOL parameter

The USEVOL parameter allows non-specific allocation requests to be directed to volumes based on use attribute (storage or private) as well as pool definition. This allows significant flexibility in volume assignment.

REJECT parameter

The REJECT parameter allows strict control of volume assignments by directing that the recommended device assignment (from OS/390) be rejected. If rejected, the data set allocation fails, and the job fails.

NVOLxxxx parameters

The NVOLxxxx parameters allow the number of volumes for a multi-volume allocation to be altered. NVOLMAX restricts the total volumes to the number of volumes in the assigned pool. NVOL and NVOLINDX restrict total volumes to the specified number for non-VSAM and VSAM data sets; both parameters are applied even if the data set is not assigned to a pool.

VFORCE parameter

The VFORCE parameter allows enforcement of standard VSAM component suffixes (DATA and INDEX). If VFORCE=YES, a DATA or INDEX suffix is added to non-JCL defined VSAM component names that do not already have that suffix. This parameter is applied regardless of pool assignment.

UNIT parameter

The UNIT parameter allows a unit name to be specified for a data set allocation. This parameter is applied regardless of pool assignment.

Warning! If the pool name (UNIT) is defined as a name that is coded in JCL, the function is active regardless of whether the mode is specified as SIMulate. All data sets that are included or not explicitly excluded from this rule for pooling are automatically allocated to this JCL-coded pool. Therefore, defining the UNIT in this manner should be used with extreme care.

VOLSEL parameter

The VOLSEL parameter specifies the type of volume selection to be performed (BESTFIT, CRITDSN, DPO, HISTDPO, MAXSPACE, or PERCENT), allowing a significant degree of control over the selection of the volume within the pool. VOLSEL is not required, and there is no default value; if not specified, OS/390 selects a volume from those in the DASDPOOL-assigned pool. The VOLSEL parameter takes effect only if at least one pool has been assigned.

USELIMIT parameter

The USELIMIT parameter specified for pools also affects pool and volume selection for non-DFSMS data sets. The USELIMIT parameter attempts to ensure that sufficient space is left on a volume to allow existing data sets to be extended with secondary extents. USELIMIT specifies a percentage utilization that should not be exceeded for a volume; if at all possible, additional primary allocations are not made on a volume if they would cause the total allocated space to exceed the USELIMIT. USELIMIT is specified in the pool definition in SMPPOOLxx and applied during DASDPOOL VOLSEL processing and SPACVOLA processing.

Note: If VOLSEL=CRITDSN is used, USELIMIT is ignored.

The USELIMIT parameter on a pool will not prevent a pool assignment, even if a volume within the USELIMIT percentage cannot be found. In this case, the last volume found which would satisfy the primary allocation will be selected.

Although multiple pools may be specified on a rule (for example, SET POOL=(TEST,WORK,TEMP)), USELIMIT is applied only to a single pool at a time; that is, the first pool is examined for space, including USELIMIT considerations; if no allocation can be made to the first pool, the second pool is examined, and so on.

Messages

The DASDPOOL function issues one or more of the following messages:

SVM3310I *job,step,dd,dsn* UNITNAME CHANGED FROM *old unit* TO *new unit*

SVM3311I *job,step,dd,dsn* REJECT = *yes/no* SET

SVM3312I *job,step,dd,dsn* USEVOL = *volume* SET, PRIVATE VOLUMES WILL BE ELIGIBLE

SVM3319I *job,step,dd,dsn* NONSPECIFIC REQUEST ASSIGNED

SVM3320I *job,step,dd,dsn* SUPVOL=NO CODED - NO FURTHER DASDPOOL FUNCTIONS WILL BE PROCESSED

SVM3330I *job,step,dd,dsn* MAXIMUM NUMBER OF UNITS ALLOWED WILL BE LIMITED BY THE POOL SIZE

SVM3331I *job,step,dd,dsn* NUMBER OF UNITS FOR THE COMPONENT SET TO *nn*

SVM3332I *job,step,dd,dsn* NUMBER OF UNITS SET TO *nn*

SVM3333I *job,step,dd,dsn* CANDIDATE = YES SET, POOLED VOLUMES ARE ELIGIBLE

SVM3334I *job,step,dd,dsn* CANDIDATE = NO SET, POOLED VOLUMES ARE INELIGIBLE

SVM3343W *job,step,dd,dsn* VOLUME SELECTION CRITERIA *xxxxxxx* IGNORED, NO POOLS ASSIGNED

SVM3335I *job,step,dd,dsn* VOLUME SELECTION CRITERIA SET TO *parameter*

SVM3336I *job,step,dd,dsn* VSAM SEPARATION SET TO *parameter*

SVM3337I *job,step,dd,dsn* ENFORCEMENT OF VSAM LOW-LEVEL QUALIFIER SET TO *.data or .index*

SVM3338I *job,step,dd,dsn* ELIGIBLE POOL NAMES SET TO: *pools*

SVM3339I *job,step,dd,dsn* DPO WINDOW SET TO *nnnn* SECONDS

SVM3353I *job,step,dd,dsn* CRITLIST = *crit_list*, CRITBIAS = *crit_bias*, CRITFAIL = *crit_fail*, CRITEMC = *crit_emc*

SVM3354I *job,step,dd,dsn* CRITLIST FAILED ALL ELIGIBLE VOLUMES. ALLOCATION REJECTED

SVM3361W *job,step,dd,dsnn* ERROR ON RAID API CALL. R15 = *rc*

SVM3362I *job,step,dd,dsn* RAIDDEVICE OF *raid_type* REQUIRED

SVM3390I *job,step,dd,dsn* DPO RESPONSE TIME OBJECTIVE SET TO *nnnn* MILLISECONDS

DASDPOOL issues message SVM3460I when a data set is assigned to a volume in a pool (through the VOLSEL parameter):

SVM3460I *job,step,dd,dsn vol/unit* VOL (*volume id*) SELECTED

Tip: To allow specific data sets to be allocated on a specific volume, use SUPVOL=NO in the rule list, or select those data sets in the filter list for inactive mode.

For example, to allow the storage administrator to place any data set on a specific volume, without regard to pooling, the following should be placed at the top of the rule list:

SET	SUPVOL=NO
INC	USER=STGADM/

SUPVOL=NO allows a JCL-specified volume to override the pool specification in the rule list for DASDPOOL. (Note that a pool assignment is not necessary.)

An alternate method is to place the following at the top of the filter list for DASDPOOL:

SET	MODE=INACT
INC	USER=STGADM/

Data sets for user STGADMxx will not invoke DASDPOOL, because the function's mode is set to INACTIVE for any user ID beginning with STGADM.

DASD Performance Optimization

The MAINVIEW SRM DPO facility is a specialized algorithm used during data set allocation to select a volume based on short-term historical performance data.

During I/O, the amount of time needed to complete the read or write operation is called the **response time**. The response time for a device can vary greatly due to the number of tasks competing for access to the volume and the channel path(s) to the volume. The DPO facility records device response times every 30 seconds and tables this performance data, maintaining 30 minutes worth of data in storage.

During allocation, the DPO facility scans the response time tables for the pooled devices over the indicated time interval; for example, for the last two minutes. The volume selected for allocation is the volume with the best response time. Alternatively, DPO can be used to select volumes based on response time objectives; for example, to select a volume that can provide a response time in the range of 20 to 40 milliseconds for some data sets or jobs and a volume that can provide a response time in the range of 40 to 60 milliseconds for other less response-critical data sets or jobs.

Within the MAINVIEW SRM ISPF interface, performance data can be displayed for each pool.

Starting DPO

The DPO facility is started and stopped automatically as part of the SVOS started task.

Resources

DPO maintains a table of performance data above the 16MB line in extended private storage. The table contains one entry for each DASD UCB for 30 minutes (sampled twice a minute); each entry is 16 bytes long. If the SVALLDPO task fails on start-up due to insufficient storage, the REGION JCL parameter should be increased for the loader job.

Usage

VOLSEL=DPO must be specified in the DASDPOOL rule for all data sets that are to use the DASD Performance Optimization allocation method.

The DPOWIND rule list parameter is used to specify the time interval over which the performance analysis is performed. For example, DPOWIND=120 indicates that the device with the best response time in the last 120 seconds should be selected for allocation. Since 30 minutes worth of data is retained, 1800 seconds is the logical upper limit for the DPOWIND operand.

Allocation based on response time objectives, rather than just the best response time, is specified in the DPORMIN and DPORMAX rule list parameters. These parameters set the minimum and maximum response times that are desired for the device to receive the allocation. For example, DPORMIN=20 and DPORMAX=40 will attempt to select a device that can provide a response time in the range of 20 to 40 milliseconds. If both DPORMIN and DPORMAX are zero, the best performing device is selected.

When multiple data sets are allocated in the same jobstep and some or all are using DPO allocation, the DPORSEP parameter is used to prevent all data sets from allocating to the same device, thus possibly degrading its performance significantly. The separation factor in DPORSEP is added to a previously used device's response time and the resulting figure is used in the selection process. This will artificially increase the device's response time, and cause subsequent allocations in the same job step to be allocated to another device with the desired response time characteristics.

In pools with fairly even performance statistics, the value of DPORSEP should generally be the average difference in response times on the different volumes. In pools with large differences in performance, such as pools that contain both cached and non-cached devices, allocations will concentrate on the fastest volumes, regardless of the separation factor.

In order to get a valid test of the DPO options, you should wait 30 minutes after you start DPO before you try to pool using DPO. It will take 30 minutes to collect the a full set of performance data. Any DPO-based allocations made before a full set of performance data has been collected may not reflect your actual selection algorithm.

Performance Data Display

The MAINVIEW SRM interface displays performance information for the devices in a pool. From the MPOOL view, type the **P** action line command to display performance data. The MVOLPER view (Figure 3-1) displays performance data (over the last 5 minute interval, maintained by DPO) for a selected pool, SMS pool, EMC physical device, or IBM physical device.

Figure 3-1 MVOLPER View

```

02JAN2001 13:29:57 ----- INFORMATION DISPLAY -----
COMMAND ==>
CURR WIN ==> 1          ALT WIN ==>
>W1 =MVOLPER=====SJSG=====02JAN2001==13:29:57====MVSRM=====16
C Volume UCB   Resp Queue  Pend  Disc  Conn  Busy  Serv  EXCPs # of SSI
- - - - - Addr Time  Time   Time Time Time Time Time ----- UCBs Targ
SMS011 85A3   4.43   0.00   0.17  0.00  4.26  0.03  4.26     3    1 SJSG
SMS301 853E   4.43   0.00   0.21  0.00  4.22  0.03  4.22     3    1 SJSG
SMS002 83BD   4.42   0.00   0.12  0.08  4.22  0.04  4.30     3    1 SJSG
SMS303 8540   4.39   0.00   0.21  0.00  4.18  0.03  4.18     3    1 SJSG
SMS001 8355   4.38   0.00   0.12  0.08  4.18  0.03  4.26     3    1 SJSG
SMS004 8357   0.62   0.00   0.25  0.12  0.25  0.00  0.37     1    1 SJSG
SMS21B 8524   0.50   0.00   0.25  0.00  0.25  0.00  0.25     1    1 SJSG
SMS306 8543   0.50   0.00   0.25  0.00  0.25  0.00  0.25     1    1 SJSG
SMS305 8542   0.50   0.00   0.25  0.00  0.25  0.00  0.25     1    1 SJSG
SMS304 8541   0.50   0.00   0.25  0.00  0.25  0.00  0.25     1    1 SJSG
SMS25A 8358   0.50   0.00   0.25  0.00  0.25  0.00  0.25     1    1 SJSG
SMS003 8356   0.50   0.00   0.25  0.00  0.25  0.00  0.25     1    1 SJSG
SMS302 853F   0.49   0.00   0.12  0.12  0.25  0.00  0.37     1    1 SJSG
MSG00 8576   0.37   0.00   0.12  0.00  0.25  0.00  0.25     1    1 SJSG
MSA00 85A2   0.00   0.00   0.00  0.00  0.00  0.00  0.00     0    1 SJSG
MSA01 8579   0.00   0.00   0.00  0.00  0.00  0.00  0.00     0    1 SJSG

```

Historical DASD Performance Optimization

The MAINVIEW SRM HISTDPO facility allows you to pool data sets based on performance data gathered over a period of up to two weeks. Historical performance data is obtained using Performance Collector in the MAINVIEW SRM Reporting component, which must be licensed and operational at your site.

Data is stored in a data space to decrease the access time required to obtain the performance data during allocation. The data is averaged and compared against the options you choose to determine the optimum volume.

Usage

The system parameter, HISTDAYS, specifies the number of days of data to retrieve from the Performance Collector and temporarily store in a data space. Values are 0-14 (7 or 14 recommended), which indicate the number of days of data to be retrieved. The default value is 0, which indicates that the data space is created, but no historical performance data is gathered and no HISTDPO pooling can be performed. If this value is changed, the Performance Collector (SVSGP) must be stopped and restarted. For a valid HISTDPO test, do not specify a number of HISTDAYS higher than the total number of days collected in the Performance Collector.

VOLSEL=HISTDPO must be specified in the DASDPOOL rule for all data sets that are to use the HISTDPO allocation method.

Starting time, ending time, and days of collection are specified in HDPOSTIM, HDPOETIM, and HDPODAYS respectively.

Allocation based on response time objectives, rather than just the best response time, is specified in the HDPORMIN and HDPORMAX rule list parameters. These parameters set the minimum and maximum response times that are desired for the device to receive the allocation. For example, HDPORMIN=20 and HDPORMAX=40 will attempt to select a device that can provide a response time in the range of 20 to 40 milliseconds. If both HDPORMIN and HDPORMAX are zero, the best performing device is selected.

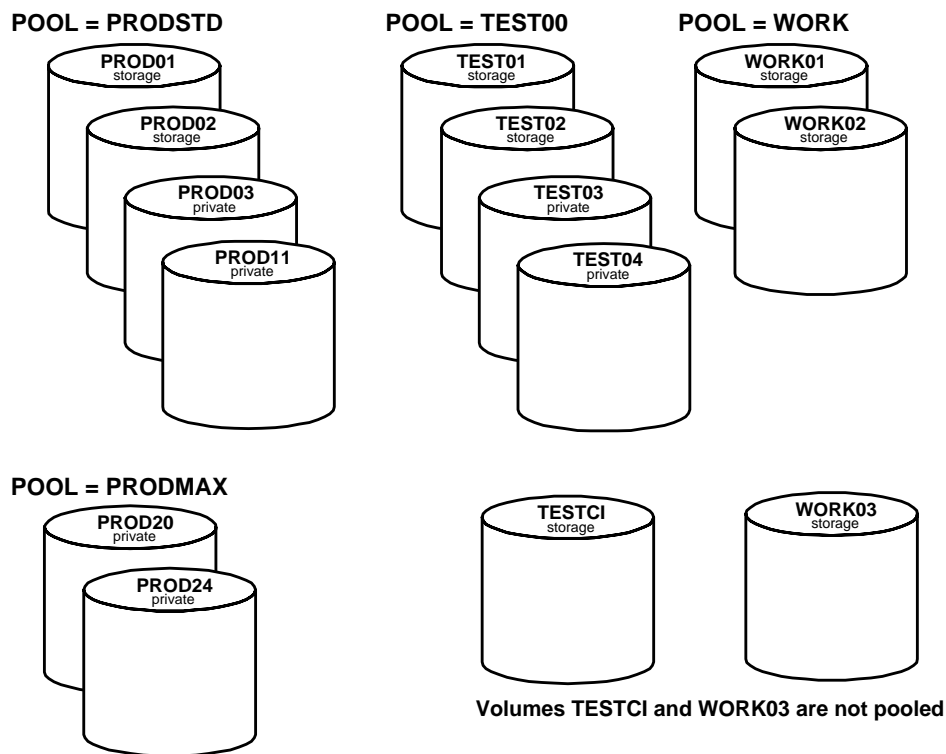
When multiple data sets are allocated in the same jobstep and some or all are using HISTDPO allocation, the HDPORSEP parameter is used to prevent all data sets from allocating to the same device, thus possibly degrading its performance significantly. The separation factor in HDPORSEP is added to a previously used device's response time and the resulting figure is used in the selection process. This will artificially increase the device's response time, and cause subsequent allocations in the same job step to be allocated to another device with the desired response time characteristics.

In pools with fairly even performance statistics, the value of HDPORSEP should generally be the average difference in response times on the different volumes. In pools with large differences in performance, such as pools that contain both cached and non-cached devices, allocations will concentrate on the fastest volumes, regardless of the separation factor.

Example

This example shows how to define pools for test, production, and work volumes, as shown in Figure 3-2.

Figure 3-2 Possible System Storage Configuration



SMMSYS10 member

```
SET  FUNC=11 POOL=11 CAL=21
      VAR=11
      CNDPREF=# SMFID=250
      MSGID=YES TRKLEN=47476 TRKCYL=15
      PASSWORD=DB6145678901E
```

The MAINVIEW SRM system is defined.

SMFUNC11 member

```
SET  NAME=DASDPOOL ACTIVE=YES
      MSG=I SMF=I
      FLST=D4 RLST=D4
      DESC='ALLOCATE DATA SETS TO POOLS'
```

The function is defined and activated.

SMPOOL11 member

SET POOLNAME=PRODSTD USELIMIT=90
VOL=PROD0/
VOL=PROD11

Pool PRODSTD includes all volumes whose names begin with PROD0 and volume PROD11. A 90% capacity limit is put on all volumes in this pool.

SET POOLNAME=PRODMAX USELIMIT=80
VOL=PROD2/

Pool PRODMAX includes two high-capacity 3390 volumes designated for very large data sets.

SET POOLNAME=TEST00
EXC VOL=TESTC/
INC VOL=TEST/

Pool TEST00 includes all volumes whose names begin with TEST, except those beginning with TESTC.

SET POOLNAME=WORK
VOL=(WORK01,WORK02)

Pool WORK includes the two named volumes.

SMVARS11 member

SET VARIABLE=SYSPGMRS
VALUE=CRMORGN/
VALUE=SSMITH/
VALUE=JAPPLEY/
VALUE=DLITCHR/
VALUE=AJOHNSN/

Define a variable that contains the user IDs of all system programmers.

SMFLSTD4 member

SET MODE=INACT
INC JOB=&SYSPGMRS DD=SYSDD/

By setting the mode to inactive, do not pool data sets with a DD name that starts with SYSDD that is in a job submitted by a system programmer. (This allows an override capability for system programmers; since pooling is not activated, the data set will be placed on the volume specified in the JCL.)

SET MODE=ACT
INC XMODE=JOB
INC XMODE=TSO

For all other data sets allocated by batch jobs or TSO sessions, pooling is active.

SMRLSTD4 member

SET SUPVOL=NO
INC DSORG=VS VOL=DV5AR/

Do not pool any VSAM data sets that request allocation on volumes whose names begin with DV5AR.

SET POOL=PRODMAX USEVOL=ALL
VOLSEL=PERCENT
EXC DSN=TEST/
INC MAXSIZE>81MB

X Pool PRODMAX is selected for all data sets with a potential maximum size greater than 81MB, except for data sets whose name begins with TEST. All volumes in the pool (storage and private) are eligible for allocation.

SET	POOL=PRODSTD USEVOL=ALL INC JOBCLASS=P INC DSN=PR/ EXC DSN=TEST/ INC LLQ=MASTER		Pool PRODSTD is selected for all data sets in jobs with a job class P, for all data sets beginning with PR, and for all data sets ending with MASTER but not beginning with TEST. All volumes in the pool (storage and private) are eligible for allocation.
SET	POOL=WORK SUPVOL=YES VOLSEL=MAXSPACE INC DSTYPE=TEMP	X	All temporary data sets are assigned to pool WORK. Requests for specific volumes are ignored.
SET	POOL=TEST00 USEVOL=PRIV INC VOL=TEST/ UNIT=TEST		Data sets allocated with UNIT=TEST and a specific volume (starting with TEST) are placed on the private volumes (TEST03, TEST04) in the TEST00 pool.
SET	POOL=TEST00 USEVOL=STOR INC UNIT=TEST		Data sets allocated with UNIT=TEST are placed on the storage volumes (TEST01, TEST02) in the TEST00 pool.
SET	POOL=(WORK,TEST00) USEVOL=PRIV INC JOBACCT3=PRIV LLQ=RESTRICT		Private volumes only in pools WORK and TEST00 receive data sets with a third job account field of PRIV and a final name qualifier of RESTRICT.
SET	POOL=(WORK,TEST) USEVOL=ALL INC DSN=/ 		All other data sets (selected by the SMFLSTD4 parameters above) are assigned to pool WORK if possible, and pool TEST if the WORK volumes are full.
SET	POOL=PRODSTD VOLSEL=DPO DPOWIND=240 DPORMIN=20 DPORMAX=30 DPORSEP=11 DPORDEF=40 INC DSN=TRAN.MSTR.COLL/ INC DSN=CICS*.REG*.LOG/	X X X	For two classes of data sets that require above-average response times, allocate to devices that over the last 4 minutes are providing response times in the range of 20 to 30 milliseconds and force separate volume allocation within the same jobstep.
SET	VFORCE=YES INC JOBCLASS=P		Ensures that all VSAM data sets created by production jobs use standard suffixes.
SET	UNIT=EMPDA INC UNIT=SYSDA		Change SYSDA allocations to EMPDA.

DSNCHECK—Enforce Data Set Naming Standards

The DSNCHECK function checks data set names for conformance to standards.

Data set name standards can significantly ease the transition to automated storage management. Standard naming conventions can embed important management information within the data set name, allowing more intelligent handling of data sets in storage management routines. The implementation of DFSMS is also easier if a standard data set naming process is in place.

DSNCHECK allows the data center to set and enforce standards for data set names. Name standards can be applied to the first eight data set name qualifiers and to the low-level qualifier. Using filter lists, rule lists, and name masking, flexible name standards can be specified and applied to resource groups selected on the basis of data set name, user name, job name, pool, volume, and so on.

Running DSNCHECK in simulation mode allows a gradual phase-in of new name standards. With normal messaging turned off (MSG=N) and SMF logging turned on (SMF=I), the administrator can preview the impact of new naming standards without any visible changes to the users. With normal messaging turned on (MSG=I), but still in simulation mode, the users are warned of impending standards implementation by the violation messages in their job logs (but without job cancellations). When an appropriate acclimatization period has passed, the function is put in active mode, and the naming standards are enforced.

System Parameters

The system parameters that affect DSNCHECK are described in Table 3-3.

Table 3-3 **System Parameters—DSNCHECK (Part 1 of 2)**

Parameter	Purpose
DASDGENR	specifies the only generic DASD names that are to be processed for specific functions if no space requirements are specified Standard pooling functionality allows the JCL UNIT parameter to specify a pool name. If this is not desired, DASDGENR can be used to intercept data set allocations without space information. Such allocations will only be considered for subsequent processing if the unit is found in an internal table (3380, 3350, and so on) or if the unit is found in the DASDGENR list. DASDGENR affects functions DASDPOOL, DSNCHECK, SETEXPDT, and FORCECAT.

Table 3-3 System Parameters—DSNCHECK (Part 2 of 2)

Parameter	Purpose
REJECT	controls termination of processing at the first or last data set rejected by REJECT=YES in DASDPOOL or DSNCHECK If termination is to take place on the first rejected data set, code REJECT=FIRST. If all data sets are to be processed before control is returned to OS/390 allocation, code REJECT=LAST.
TAPEGENR	specifies tape device generic names that some EasyPOOL functions will intercept If you want to intercept all tape requests, specify ALLTAPE as the first generic name. (However, you cannot use ALLTAPE when JCLEXT=NO.) TAPEGENR affects functions DSNCHECK, FORCECAT, SETEXPDT, and TAPEPOOL.

Rule List Parameter Quick Reference

Parameter	Description
DSNn=xxxxxxxx	Name of data set qualifier 1–8
EVENTID=xxxxx	Identifies a user event defined in an SMEVNTxx member
LLQ=xxxxxxxx	Name of low-level data set qualifier
MAXQLF=nn	Maximum number of data set name qualifiers
MINQLF=nn	Minimum number of data set name qualifiers
VCOMPLLQ=xxxxxxxx	Name of VSAM component low-level data set qualifier

Rule List Parameter Explanations**DSNn=**

Purpose: Contains a data set name qualifier; *n* is a number from 1 to 8, indicating which data set name qualifier is to be checked.

If the data set name qualifier matches this parameter, the qualifier is accepted and the next qualifier or other parameter is checked; if the name qualifier does not match, the data set name is rejected with a naming violation (message SVM0344E) and no further name qualifiers are checked.

Note that while name masking cannot normally be used in a rule list parameter, in this particular case, name masking is accepted in the rule SET statement.

Syntax: DSN*n*=xxxxxxxx

where *n* is a number in the range 1–8 and xxxxxxxx is any character string valid as a data set name qualifier.

Default: None

EVENTID=

Purpose: Specifies the identifier assigned to a user event in SMEVNTxx. This parameter will cause an event to be generated from this function.

Note: If EVENTID= is used on an FLST SET statement with MODE=INACT, the event will still be issued.

Syntax: EVENTID=xxxxxx

where xxxxx is the 5-character string specified on the EVNTID parameter in SMEVNTxx.

Default: None

LLQ=

Purpose: Contains a low-level data set name qualifier. If the last qualifier in the data set name matches this parameter, the qualifier is accepted; if the name qualifier does not match, the data set name is rejected with a naming violation (message SVM0344E).

Note that while MAINVIEW SRM name masking cannot normally be used in a rule list parameter, in this particular case, name masking is accepted in the rule SET statement.

Syntax: LLQ=xxxxxxxx

where xxxxxxxx is any character string valid as a data set name qualifier.

Default: None

MAXQLF=

Purpose: Specifies the maximum number of name qualifiers a data set is allowed. If the data set name contains more qualifiers than this number, the data set name is rejected with message SVM0342E.

Syntax: MAXQLF=nn

where nn is a number in the range 1–99.

Default: None

MINQLF=

Purpose: Specifies the minimum number of name qualifiers a data set is required to contain. If the data set name has fewer qualifiers than this number, the data set name is rejected with message SVM0341E.

Syntax: MINQLF=*nn*

where *nn* is a number in the range 1–99.

Default: None

VCOMPLLQ=

Purpose: Contains a low-level data set name qualifier for a VSAM data component. If the last qualifier in the VSAM component matches this parameter, the qualifier is accepted; if the name qualifier does not match, the data set name is rejected with a naming violation (message SVM0344E).

Note that while MAINVIEW SRM name masking cannot normally be used in a rule list parameter, in this particular case, name masking is accepted in the rule SET statement.

Syntax: VCOMPLLQ=*xxxxxxxx*

where *xxxxxxxx* is a valid data set name qualifier 1–8 characters long. It will contain blanks for the cluster.

Default: None

Parameters Not Supported

The following filter list parameters are not supported for DSNCHECK:

- GDGVER
- VALUE

Usage Notes

DSNCHECK provides services for VSAM and non-VSAM data sets. It operates on non-VSAM data sets at initial allocation and at data set rename. For VSAM data sets, it operates on DEFINE and ALTER operations for both the data and index components of the data set.

Note that DSNCHECK does not operate on:

- Data sets with only one qualifier in the name
- Temporary data sets

DSNCHECK acts on the first eight qualifiers in the data set name and on the low-level qualifier, regardless of how many qualifiers precede it.

Contrary to other rule list parameters, the DSNCHECK rule list parameters DSNn and LLQ can contain name masks. At execution time, these two parameters, as well as MINQLF and MAXQLF, are compared to their respective data set name components. If there is a match, the data set name qualifier is accepted; if there is not a match, a naming violation is reported. *Once a qualifier has failed a name check, no further qualifiers in the data set name are checked for compliance.* The MINQLF and MAXQLF parameters are checked first, followed by the DSNn parameters in numerical order, and then the LLQ parameter.

When DSNCHECK is activated for a rename operation, all DSN filter list parameters (DSN1-8, HQL, LLQ) contain values from the new data set name. A rename condition is identified by a value of RENAME in the ENVIR filter list parameter.

Naming standards violations are identified by one of three MAINVIEW SRM messages. If the function is running in active mode, the job is caused to fail with a JCL error (a VSAM IDCAMS DEFINE will fail with a VSAM catalog return code of 68 - unable to allocate space on user volume). For example:

```
06.24.50 JOB04488 SVM0344E EMPCRM ,DD2,EMPCRM.V3R3M2.PROPLIB QUALIFIER(2) VIOLATES NAMING
STANDARDS
06.24.50 JOB04488 IEF452I EMPCRM - JOB NOT RUN - JCL ERROR
```

A name violation on a data set rename will be accompanied by OS/390 message “IEC614I - RENAME FAILED ...” for non-DFSMS data sets and by OS/390 message IEC331I for DFSMS data sets. Error messages for IEHPROGM or IDCAMS will also be produced with the SYSOUT for those utilities.

The VFORCE parameter of the DASDPOOL function also provides data set name services; VFORCE=YES adds standard component suffixes (DATA, INDEX) to VSAM file components.

Note that the DSNCHECK function must be specified in the SMFUNC_{xx} member to be available and must be set active to provide enforcement services.

Naming standards and DFSMS are discussed in “Developing Naming Conventions” in the IBM® manual *MVS/ESA Storage Management Library: Managing Data*.

Example

This example shows how to:

- require all load module partitioned data sets to end with LLIB, and all source partitioned data sets to end with SLIB
- all partitioned data sets must have at least three qualifiers in their names
- run in simulate mode to assess the impact of this change

SMFUNC11 member

```
SET  NAME=DSNCHECK ACTIVE=YES
      MSG=I SMF=N
      FLST=17 RLST=18
      DESC='DATA SET NAME CHECKING'
```

The function is defined and activated. No SMF records will be produced.

SMFLST17 member

```
SET  MODE=SIM
      EXC HLQ=SYS*
      EXC HLQ=IS*
      INC DSORG=PO
```

Data set name checking for resources selected by this filter list will be in simulate mode only. All partitioned data sets are selected, except those beginning with SYS and IS.

SMRLST18 member

```
SET  LLQ=*LLIB MINQLF=3
      INC RECFM=U
```

For all load libraries (RECFM=U), the low-level name qualifier must end with LLIB. Also, the data set name must contain at least three name qualifiers.

```
SET  LLQ=*SLIB MINQLF=3
      INC LRECL=80
```

For all source libraries (record length of 80), the low-level name qualifier must end in SLIB, and the name must have at least three qualifiers.

Example

This example shows how to require the second qualifier of all data sets except system-level data sets (contained in pools SYSM1 and SYSRES) to be one of the following division codes: ATW, MKE, POS, HDA, CTY, FRP, with the following conditions:

- Data sets allocated by production jobs must have at least four name qualifiers, must have a valid DP system ID in the high-level qualifier, and must have a low-level qualifier beginning with P for non-VSAM data sets and PV for VSAM data sets.
- All other data sets must have at least three name qualifiers.
- PDSs must have LIB in the low-level qualifier.
- Data sets created by systems programmers are exempt from all naming standards, as are data sets residing on DB2® packs.

SMPOOLB2 member

SET	POOLNAME=SYSM1 TYPE=DASD INC VOL=SYS/ INC VOL=(BTD001,NRID0A,NRID33)	Define the SYSM1 pool to contain all volumes whose names start with SYS and three other volumes.
SET	POOLNAME=SYSRES INC VOL=*RES	Define the SYSRES pool to contain sysres volumes.
SET	POOLNAME=DB2POOL INC VOL=DB2/	Define the DB2POOL pool to contain all volumes whose names start with DB2.

SMFUNCB2 member

SET	NAME=DSNCHECK ACTIVE=YES MSG=I SMF=I FLST=B1 RLST=B1 DESC='ENFORCE NAME STANDARDS'	The function is defined and activated.
-----	---	--

SMVARSA1 member

SET	VAR=DIV VALUE=ATW VALUE=MKE VALUE=POS VALUE=HDA VALUE=CTY VALUE=FRP	Define a variable to hold all the division names.
SET	VAR=SYSPPOOL VALUE=SYSM1 VALUE=SYSRES	Define a variable to hold the two system pools.

Chapter 3 EasyPOOL Functions 3-37

FDRASIST—Pool Data Sets Recalled With FDR/ABR

The FDRASIST function assigns non-VSAM data sets being recalled by FDR/ABR to pools of DASD volumes. FDRASIST does not work for VSAM data sets.

Considerations in pooling during FDR recall include:

- use of storage versus private volumes
- specification of multiple pools, allowing overflow assignment if one or more pools are full
- criteria for selection of the *best* volume in a pool

Along with the DASDPOOL and HSMRECAL functions, FDRASIST allows the data center to set and enforce standards for data sets' volume residency. Data sets being recalled by FDR can be directed to MAINVIEW SRM-defined pools on the basis of a variety of job and data set properties; for more flexibility in space utilization and management, multiple pools can be assigned, and multiple methods of volume selection can be used.

Rule List Parameter Quick Reference

Parameter	Description
CRITBIAS=	Defines the number of data sets that can reside on the volume before the current allocation
CRITEMC= YES/NO	Specifies whether the volume meeting CRITDSN criteria includes EMC physical volumes
CRITFAIL= YES/NO	Defines the allocation process if a volume meeting the criteria cannot be found
CRITLIST=xxxxxxx	Name of the table that contains allocation volumes
EVENTID=xxxxx	Identifies a user event defined in an SMEVNTxx member
POOL=xxxxxxx	Name of a pool 1–8 characters
USEVOL=STOR/PRIV/ALL	Type of volume
VOLSEL=BESTFIT/PERCENT/MAXSPACE	Method of volume selection from a pool

Rule List Parameter Explanations**CRITBIAS=**

Purpose: Defines the number of data sets that can reside on the volume before the current allocation.

Syntax: CRITBIAS=*nnn*
where *nnn* is a number from 1 to 999

Default: None

Note: This parameter works only in conjunction with
VOLSEL=CRITDSN

CRITEMC=

Purpose: Specifies whether the volume meeting CRITDSN criteria includes EMC physical volumes. If yes, data sets specified with CRITDSN should not reside on the same EMC Physical volume nor the same MVS Logical volume. If no, data sets specified with CRITDSN may reside on the same EMC physical volume. Default is no.

Syntax: CRITEMC=*YES/NO*

Default: CRITEMC=NO

CRITFAIL=

Purpose: Defines the allocation process if a volume meeting the criteria cannot be found. If YES, allocation fails; if NO, allocation is made to the best available volume. Like USELIM, if an acceptable volume cannot be found in the first pool, the best volume from the first pool is saved and the system tries the next pool(s). If no acceptable volume is found, CRITFAIL is processed on the saved volume.

Syntax: CRITFAIL=*YES/NO*

Default: CRITFAIL=NO

Note: This parameter works only in conjunction with
VOLSEL=CRITDSN

CRITLIST=

Purpose: The table that contains allocation volumes.

Syntax: CRITLIST=xxxxxxxx

Default: None

Note: This parameter works only in conjunction with
VOLSEL=CRITDSN

Warning! CRITDSN is resource intensive and should be used only for a small list of critical data sets. It should not be used without considering the impact on the system.

EVENTID==

Purpose: Specifies the identifier assigned to a user event in SMEVNTxx. This parameter will cause an event to be generated from this function.

Note: If EVENTID= is used on an FLST SET statement with
MODE=INACT, the event will still be issued.

Syntax: EVENTID=xxxxx

where xxxxx is the 5-character string specified on the EVNTID parameter in SMEVNTxx.

Default: None

POOL=

Purpose: Specifies the name of a pool. Up to 15 values of this parameter can be specified within parentheses when used as a rule list parameter on a SET statement.

Syntax: POOL=xxxxxxxx or POOL=(xxxxxxxx,xxxxxxxx,...)

where xxxxxxxx is a valid pool name 1–8 characters long. If the poolname is not defined, the refresh or start-up will fail. Up to 15 pool names can be specified in parentheses.

Default: None

USEVOL=

Purpose: Specifies the type of volume (storage, private, or all) that satisfies non-specific allocation requests.

Syntax: USEVOL=xxxx

where xxxx is a value from the following list:

STOR Storage mounted volume
 PRIV Privately mounted volume
 ALL Volume of any mount type

Default: USEVOL=ALL

VOLSEL=

Purpose: Specifies that volume selection within a pool should be based on current volume space availability.

Syntax: VOLSEL=BESTFIT/PERCENT/MAXSPACE

BESTFIT specifies that the volume with the smallest contiguous extent that satisfies the primary allocation should be selected

MAXSPACE specifies that the volume with the largest single contiguous extent should be selected

PERCENT specifies that the volume with the largest amount of free space should be selected

Default: VOLSEL=BESTFIT

Parameters Not Supported

The following filter list parameters are not supported for FDRASIST:

ACF2USER	BUFSP	CATALOG
CISIZE	CURDAY	CURSPACE
CURTIME	DD	DEVTYPE
DISP1	DISP2	DISP3
DSNTYPE	DYNALLOC	ENVIR
ERASE	FILESEQ	GDGVER
IMBED	JOBSDAY	JOBSTIME
LABELTYP	OLDDSN	OLDHLQ

OWNER	RELEASE	REPL
REUSE	UNIT	VALUE

Usage Notes

For FDRASIST to gain control, the volume requested on the NVOL parameter of the FDR SELECT statement must be rejected. This can be accomplished by specifying a volume in the FDR SELECT statement that does not exist in the pool assigned to the data set by the FDRASIST rule list. (The rejection also occurs if the NVOL-specified volume is full.)

Note: The NVOL-specified volume must exist and must be online.

FDRASIST will not work on VSAM data sets. For non-VSAM data sets, the FDR option ALCRSTIN in FDR releases 5.3 and higher, or the POOLDASD option in release 5.2, must be enabled. Refer to the FDR manual for additional information about this interface.

The module PLDRSTIN from the MAINVIEW SRM load library must be available at run time to the FDR job (for instance, in a steplib or the linklist).

The POOL parameter allows specification of one or more pools; pools are searched in the order specified. Note that any USELIMIT defined for a pool will be taken into account in selecting the pool and assigning a volume. However, USELIMIT will not prevent a pool assignment. If a pool that satisfies the uselimit requirement cannot be found, the last pool in the POOL parameter will be chosen.

The VOLSEL parameter specifies the type of volume selection to be performed, allowing a degree of control over the selection of a specific volume from a pool. If VOLSEL is not specified, the BESTFIT method will be used. Note that the MAXSPACE and PERCENT methods will always select a volume in a pool; however, if VOLSEL=BESTFIT cannot be satisfied, the pool/volume assignment will fail.

FDR does not interface with EasyPOOL for VSAM data sets. For VSAM data sets, DASDPOOL is used to assign data sets to pools of DASD volumes.

Example

This example shows you how to pool FDR/ABR recalls.

SMFUNC03 member

```
SET  NAME=FDRASIST ACTIVE=YES
      MSG=I SMF-I
      FLST=F1 RLST=F2
      DESC='POOL ON FDR RECALL'
```

SMPOOL10 member

```
SET  POOLNAME=FDR1
      VOL=WORK/

SET  POOLNAME=FDR2
      VOL=EMP/
```

SMFLSTF1 member

```
SET  MODE=ACT
      INC PGM=FDR*
```

FDRASIST is active for any program whose name begins with FDR.

SMRLSTF2 member

```
SET  POOL=FDR1
      INC PGM=FDR* HLQ=EMP

SET  POOL=FDR2
      VOLSEL=PERCENT
      INC PGM=FDR* HLQ=BBSD
```

Data sets with a high-level qualifier of EMP being processed by any program whose name begins with FDR will be recalled to pool FDR1, while data sets with a high-level qualifier of BBSD will be recalled to the volume with the largest single extent in pool FDR2.

FORCECAT—Force Catalog Disposition for New Data Sets

The FORCECAT function forces the normal termination disposition to CATLG for new data sets.

Effective storage management starts with properly tracking the location of data sets. Uncataloged data sets must be manually tracked or be lost, taking up valuable space on storage devices. Locating those uncataloged data sets may require a lengthy search of all volumes on the system. DFSMS manages only cataloged data sets. Data set cataloging is controlled by the JCL DD parameter DISP, but for non-DFSMS-managed data sets, OS/390 provides no way to require that the appropriate value of DISP is used.

For new data set allocations, FORCECAT forces the data set to be cataloged, regardless of the value of the DISP parameter in the JCL.

System Parameters

The system parameters that affect FORCECAT are described in Table 3-4.

Table 3-4 System Parameters—FORCECAT

Parameter	Purpose
DASDGENR	Specifies the only generic DASD names that are to be processed for specific functions if no space requirements are specified. Standard pooling functionality allows the JCL UNIT parameter to specify a pool name. If this is not desired, DASDGENR can be used to intercept data set allocations without space information. Such allocations will only be considered for subsequent processing if the unit is found in an internal table (3380, 3350, and so on) or if the unit is found in the DASDGENR list. DASDGENR affects functions DASDPOOL, DSNCHECK, SETEXPDT, and FORCECAT.
TAPEGENR	Specifies tape device generic names that some EasyPOOL functions will intercept. If you want to intercept all tape requests, specify ALLTAPE as the first generic name. (However, you cannot use ALLTAPE when JCLEXT=NO.) TAPEGENR affects functions DSNCHECK, FORCECAT, SETEXPDT, and TAPEPOOL.

Rule List Parameters

SET Statement (in member SMRLST_{xx})

Parameter	Description
EVENTID=xxxxx	Identifies a user event defined in an SMEVNT _{xx} member

EVENTID=

Purpose: Specifies the identifier assigned to a user event in SMEVNTxx. This parameter will cause an event to be generated from this function.

Note: If EVENTID= is used on an FLST SET statement with MODE=INACT, the event will still be issued.

Syntax: EVENTID=xxxxxx

where xxxxx is the 5-character string specified on the EVNTID parameter in SMEVNTxx.

Default: None

Parameters Not Supported

The following filter list parameters are not supported for FORCECAT:

- GDGVER
- VALUE

Usage Notes

FORCECAT operates on any non-VSAM data set being created, DASD or tape, whose normal termination disposition is KEEP, for example, DISP=(xxx,KEEP). FORCECAT does not operate on DFSMS-managed data sets because DFSMS ensures that the data set is cataloged.

FORCECAT can be used in simulation mode to identify the new catalogs and aliases that must be established.

With the exception of the EVENTID rule list parameter, all resource selection must be done through the FLST.

Note that the FORCECAT function must be specified in the SMFUNCxx member to be available and must be set active to provide services.

FORCECAT generates message SVM3445I when it changes the normal termination disposition from KEEP to CATLG:

SVM3445I *job,step,dd,dsn* CATALOGING FORCED

Examples

The following example shows how to force cataloging for all data sets except those on volume NOCAT or tape pool EXPORT.

SMFUNCxx member

SET	NAME=FORCECAT ACTIVE=YES MSG=I SMF=I FLST=FC DESC='FORCE DATA SET CATALOGING'	The function is defined and activated. There is no RLST specification.
-----	---	--

SMFLSTFC member

SET	MODE=ACT EXC VOL=NOCAT EXC POOL=EXPORT INC DSN=/ 	This filter list specification is active. All data sets except those on NOCAT or pool EXPORT are selected.
-----	---	--

The following example shows how to force data set cataloging for medium and large data sets created in batch and TSO jobs.

SMFUNCxx member

SET	NAME=FORCECAT ACTIVE=YES MSG=I SMF=I FLST=48 DESC='FORCE DATA SET CATALOGING'	The function is defined and activated.
-----	--	--

SMFLST48 member

SET	MODE=ACT EXC MAXSIZE<5MB INC XMODE=JOB INC XMODE=TSO	This filter list specification is active. It selects jobs that are run in batch or from TSO which have a potential maximum size greater than 5 megabytes.
-----	---	---

MODDELET—Remove Specific Volume for MOD-DELETE Data Set Deletion

The MODDELET function removes a specific volume assignment on a data set deleted by DISP=(MOD,DELETE,DELETE).

A convenient and widely used method of deleting possibly-nonexistent data sets is specification of DISP=(MOD,DELETE,DELETE) in JCL. If the data set exists, it is found in the catalog and deleted. If the data set does not exist, it is created and then deleted. In either case, a data set which may or may not exist will be deleted, with no JCL errors. In some situations, however, a specific volume assignment is included in the MOD-DELETE job step. With MAINVIEW SRM pooling active, the data set to be deleted may not be on that volume, resulting in the failure of the data set deletion.

The MODDELET function removes the specific volume assignment from job steps using the MOD-DELETE data set deletion method, allowing the operation to proceed normally.

Rule List Parameters

SET Statement (in member SMRLSTxx)

Parameter	Description
EVENTID=xxxxx	Identifies a user event defined in an SMEVNTxx member

EVENTID=

Purpose: Specifies the identifier assigned to a user event in SMEVNTxx. This parameter will cause an event to be generated from this function.

Note: If EVENTID= is used on an FLST SET statement with MODE=INACT, the event will still be issued.

Syntax: EVENTID=xxxxx

where xxxxx is the 5-character string specified on the EVNTID parameter in SMEVNTxx.

Default: None

Parameters Not Supported

The following filter list parameters are not supported for MODDELET:

- GDGVER
- VALUE

Usage Notes

MODDELET affects only data sets in job steps where the executing program is IEFBR14 and DISP=(MOD,DELETE,DELETE) is specified. Any specific volume assignment (including multiple volumes) is removed from the allocation; in effect, VOL=SER=(volid,volid,. . .) is deleted from the JCL. The data set is then accessed from the system catalog.

MODDELET does not affect DFSMS-managed data sets or data sets that are allocated to VIO.

MODDELET generates message SVM3482I when it processes a MOD-DELETE action, including those without a specific volume assignment:

SVM3482I *job,step,dd,dsn* HANDLED BY MODDELET

Example

The following example shows to to remove specific volume assignments from MOD-DELETE data set deletions.

SMFUNCxx member

```
SET    NAME=MODDELET ACTIVE=YES
        MSG=I SMF=I
        FLST=M4
        DESC='PROCESS DISP=(MOD,DELETE)'
```

The function is defined and activated. There is no RLST specification.

SMFLSTM4 member

```
SET    MODE=ACT
        INC JOBCLASS=P
```

This filter list specification is active. All data sets in a production job class are selected for processing.

OPENEMPT—Initialize Empty Data Sets

The OPENEMPT function initializes empty data sets at job step termination.

Data sets that are allocated but never opened are a potential hazard to subsequent jobsteps and jobs. OS/390 does not initialize a data set at allocation; a new data set is initialized only when it is opened. If an uninitialized data set is later opened by another program, one of three occurrences will happen:

- The job will receive an I/O error because an end-of-file mark has not been written.
- The job will read a record left on the volume from a prior data set and possibly process it as normal data.
- The System-Determined Blocksize (SDB) may not be driven, resulting in an S001 abend.

OPENEMPT can intercept and prevent most uninitialized data set problems by initializing empty data sets at job step termination.

Rule List Parameters

SET Statement (in member SMRLSTxx)

Parameter	Description
EVENTID=xxxxx	Identifies a user event defined in an SMEVNTxx member

EVENTID=

Purpose: Specifies the identifier assigned to a user event in SMEVNTxx. This parameter will cause an event to be generated from this function.

Note: If EVENTID= is used on an FLST SET statement with MODE=INACT, the event will still be issued.

Syntax: EVENTID=xxxxx

where xxxxx is the 5-character string specified on the EVNTID parameter in SMEVNTxx.

Default: None

Parameters Not Supported

The following filter list parameters are not supported for OPENEMPT:

BUFSP	CATALOG	CISIZE
CURDAY	CURSPACE	CURTIME
ERASE	EXTENT	GDGVER
IMBED	JOBSDAY	JOBSTIME
LABELTYP	OWNER	REPL
REUSE	VALUE	

Usage Notes

OPENEMPT provides services for new non-VSAM data sets other than partitioned data sets. OPENEMPT obtains control at the termination of the jobstep and checks all data sets for an uninitialized condition. If found, the uninitialized data sets are opened and closed to initialize them.

An additional benefit of OPENEMPT is that the initialized data set can be processed by the SPACRLSE function to release allocated unused space. If the data set remained uninitialized, it would retain its full allocated space.

OPENEMPT ignores the data set in the following circumstances:

- The step is terminating normally, and the normal termination disposition is delete: DISP=(NEW,DELETE)
- The step is terminating abnormally, and the abnormal termination disposition is delete; DISP=(NEW,anything,DELETE)
- The data set has no volumes assigned
- The data set is a PDS
- The data set is subsequent to the first data set in a concatenation of data sets (only the first data set is processed due to OS/390 architectural restrictions)
- The data set is allocated to VIO
- No DCB information specified; DCB information uses a referback (DCB=*.STEP.DD)

Messages

The messages issued by OPENEMPT are:

```
SVM0500I  dsn INITIALIZED
SVM0501I  dsn NOT INITIALIZED. REASON = reason
SVM0500I  job,step,dd,dsn INITIALIZED
```

OPENEMPT found the indicated data set had not been initialized. The data set was opened and closed by OPENEMPT to set the end-of-file mark.

```
SVM0501I  job,step,dd,dsn NOT INITIALIZED, REASON=RECFM
```

The indicated data set could not be initialized because record format information was not available; for example:

```
SVM0501I  EMPCRM,STEP10,DD2,EMPCRM.OPENEMPT.TEST NOT
INITIALIZED,REASON=RECFM
```

```
//EMPCRM JOB (3500), 'MORRIS',MSGCLASS=R,CLASS=F
//STEP10 EXEC PGM=IEFBR14
//DD2 DD DSN=EMPCRM.OPENEMPT.TEST,
// DISP=(NEW,CATLG,DELETE),
// SPACE=(TRK,(1,1)),UNIT=SYSDA
```

```
SVM0501I  job,step,dd,dsn NOT INITIALIZED, REASON=DDNAME
```

The indicated data set could not be initialized because the ddname was a duplicate of another; for example:

```
SVM0500I  EMPHRFX,STEP10,DD2,EMPHRF.OPENEMPT.TEST INITIALIZED
SVM0501I  EMPHRFX,STEP10,DD2,EMPHRF.OPENEMPT.TEST2 NOT
INITIALIZED,REASON=DDNAME
```

```
//EMPHRFX JOB (3500), 'FITZGERALD',MSGCLASS=R,CLASS=F
//STEP10 EXEC PGM=IEFBR14
//DD2 DD DSN=EMPHRF.OPENEMPT.TEST,
// DISP=(NEW,CATLG,DELETE),
// SPACE=(TRK,(1,1)),UNIT=SYSDA,
// DCB=(LRECL=80,BLKSIZE=16000,RECFM=FB)
//DD2 DD DSN=EMPHRF.OPENEMPT.TEST2,
// DISP=(NEW,CATLG,DELETE),
// SPACE=(TRK,(1,1)),UNIT=SYSDA,
// DCB=(LRECL=80,BLKSIZE=16000,RECFM=FB)
```

```
SVM0501I  job,step,dd,dsn NOT INITIALIZED, REASON=OPEN
```

The indicated data set could not be initialized because it was already open. This normally indicates the program terminated without closing the data set.

Note that the OPENEMPT function must be specified in the SMFUNCxx member to be available and must be set active to provide services.

Example

This example shows how to use OPENEMPT on all data sets other than those in programs under development.

SMFUNC11 member

SET	NAME=OPENEMPT	ACTIVE=YES	The function is defined and activated. Note that there is no RLST parameter.
	MSG=I	SMF=I	
	FLST=03		
	DESC='OPEN EMPTY DATA SETS'		

SMVARS02 member

SET	VARIABLE=TESTPGMS	Set a variable to identify test jobs by program name. Any job that begins or ends with T, or that begins with DBT is considered to be a test job.
	VALUE=T*	
	VALUE=*T	
	VALUE=DBT*	
SET	VARIABLE=TESTJOBS	Set a variable to identify test jobs by job class.
	VALUE=T	
	VALUE=D	
	VALUE=E	

SMFLST03 member

SET	MODE=ACT	Select resources for OPENEMPT based on program name or job class.
INC	PGM=&TESTPGMS	
INC	JOBCLASS=&TESTJOBS	

OPTBLKSZ—Set Optimum Blocksize

The OPTBLKSZ function forces an optimum blocksize for new data set allocations on DASD and tape. With the use of optional parameters, the BLKSIZE may also be changed on OLD or INPUT data sets.

The blocksize of a data set has a direct effect on the storage efficiency of DASD devices. Generally, the larger the blocksize, the more records fit on a track. This not only achieves better utilization of disk space, it also reduces the number of I/Os required to read the data set. Although very small data sets may not benefit significantly from the denser record storage, the minimum allocation by OS/390 is a single track; therefore, it generally is more effective to allocate larger blocksizes. The only disadvantage is that larger blocksizes require larger buffers in memory. The following table shows the effect of different blocksizes on DASD capacity.

Record Density, 3380 DASD, 80-byte records			
Blocksize	Records/track	Records/cylinder	Percent space used ^a
4240	530	7950	90.0
6320	553	8295	93.7
23440	586	8790	98.9

^a Source: IBM 3380 Direct Access Storage: Reference Summary

OPTBLKSZ allows the data center to set and enforce standard data set blocksizes. Using rule lists, different blocksizes can be applied to different groups of data sets.

System Parameters

The system parameters that affect OPTBLKSZ are described in Table 3-5.

Table 3-5 System Parameters—OPTBLKSZ (Part 1 of 2)

Parameter	Purpose
BLKINPUT	Changes block size for input data sets.
BLKOLDSR	Changes blocksize for output data sets opened with disposition of old or shared.
TRKCYL	Specifies the number of tracks per cylinder for the default device type. The value specified for 3380/3390/9345 devices should be fifteen. (Note that this specification is the same as the SCDS base configuration DEFINE under ISMF for DFSMS.) TRKCYL and TRKLEN are used by the DASDPOOL function to convert allocations in tracks or cylinders to megabytes for volume selection based on available space; for example, VOLSEL=BESTFIT. The information specified on these two parameters should reflect the devices that are most prevalent in your environment.

Table 3-5 System Parameters—OPTBLKSZ (Part 2 of 2)

Parameter	Purpose
TRKLEN	<p>Specifies the number of bytes per track for the default device type. Valid values are: 3380 – 47,476 3390 – 56,664 9345 – 46,456</p> <p>Note that this specification is the same as the SCDS base configuration DEFINE under ISMF for DFSMS.)</p> <p>TRKCYL and TRKLEN are used by the DASDPOOL function to convert allocations in tracks or cylinders to megabytes for volume selection based on available space; for example, VOLSEL=BESTFIT. The information specified on these two parameters should reflect the devices that are most prevalent in your environment.</p>

Rule List Parameters

SET Statement (in member SMRLSTxx)

Parameter	Description
BLKSIZE=nnnnn	A blocksize in the range 0–32,760
EVENTID=xxxxx	Identifies a user event defined in an SMEVNTxx member
FORCE=YES/NO	Overrides program specified blocksize

Rule List Parameter Explanations

BLKSIZE=

Purpose: Specifies a blocksize to be assigned to the new data set. A specification of zero directs the use of a system-assigned blocksize. This is a required parameter; no processing occurs if a BLKSIZE is not specified.

Syntax: BLKSIZE=nnnnn

where nnnnn is a number in the range 0–32,760.

Default: None

EVENTID=

Purpose: Specifies the identifier assigned to a user event in SMEVNTxx. This parameter will cause an event to be generated from this function.

Note: If EVENTID= is used on an FLST SET statement with MODE=INACT, the event will still be issued.

Syntax: EVENTID=xxxxx

where *xxxxx* is the 5-character string specified on the EVNTID parameter in SMEVNT_{xx}.

Default: None

FORCE=

Purpose: Specifies whether any program specified blocksize should be overridden.

Syntax: FORCE=*YES/NO*

If YES, the value specified will be used to override a program specified blocksize; if NO, a program specified blocksize will NOT be overridden.

Default: FORCE=NO

Parameters Not Supported

The following filter list parameters are not supported for OPTBLKSZ:

- GDGVER
- VALUE

Usage Notes

OPTBLKSZ provides services for both new and old DASD and tape data sets with record formats of undefined, variable, fixed, or fixed blocked. *It does not operate on ISAM, direct, or VSAM data sets.* It will override blocksize assigned in program DCBs if FORCE=YES is specified. The data set allocation *must* be specified with both record length and a blocked record format (FB, VB, FBA, and so on); without these two pieces of information, a blocksize is not assigned.

A BLKSIZE rule list parameter is required; if not specified in a rule, the data set selected for processing is ignored.

If the LRECL of the selected data set is zero, or larger than the rule-specified blocksize (SET BLKSIZE=), OPTBLKSZ takes no action. Otherwise, a new blocksize is assigned. For fixed and fixed blocked data sets, the rule-specified blocksize is adjusted to the next lower multiple of the LRECL; for variable and undefined data sets, the rule-specified blocksize is used as is.

Note that a blocksize of zero can be specified to request a system-assigned blocksize; in this case, OPTBLKSZ sets the BLKSIZE to 0. OS/390 System-Determined Blocksize (SDB) must be driven to reset the BLKSIZE to the optimum value. For SDB to reset the value, the data set must be opened in the same jobstep in which it is allocated (see “OPENEMPT—Initialize Empty Data Sets” on page 3-49).

Messages

The message issued by OPTBLKSZ when a blocksize is changed is:

```
SVM3347I  job,step,dd,dsn BLKSIZE CHANGED nnnnn/nnnnn
```

The first *nnnnn* gives the original blocksize specified in the JCL; the second *nnnnn* gives the blocksize set by OPTBLKSZ.

The message issued by OPTBLKSZ when a specific blocksize is requested by program name and FORCE=YES is coded is:

```
SVM3351I  job,step,dd,dsn BLKSIZE nnnn IN PGM=name REMOVED
```

Note that the OPTBLKSZ function must be specified in the SMFUNCxx member to be available and must be set active to provide service.

Example

This examples shows you how to:

- specify blocksize for all data sets except certain system-level PDSs
- run in simulate mode for data sets in jobs with BLKEXEMPT in the third job account field

SMFUNC73 member

```
SET  NAME=OPTBLKSZ ACTIVE=YES  
      MSG=I SMF=N  
      FLST=7D RLST=7E  
      DESC='ASSIGN OPTIMUM BLOCKSIZE'
```

The function is defined and activated. No SMF records will be produced.

SMFLST7D member

```
SET  MODE=SIM  
      INC JOBACCT3=BLKEXEMPT
```

The function is to run in simulate mode for all data sets in jobs with BLKEXEMPT in the third job accounting field.

```
SET  MODE=ACT  
      EXC HLQ=SYS* DSORG=PO RECFM=U  
      INC DSN=/'
```

The function is to run in active mode for all selected resources. All data sets are selected, except system-level undefined partitioned data sets.

SMRLST7E member

SET	BLKSIZE=3861 INC DSORG=PO LLQ=JCL/	Specify a small blocksize for JCL PDSs, since they consist of many members with 50 records or less.
SET	BLKSIZE=23440 INC MAXSIZE>50MB DEVTYPE=3380 INC MAXSIZE>50MB DEVTYPE=3390	Specify half-track blocking for large data sets on all 3380 and 3390 devices.
SET	BLKSIZE=4629 INC MAXSIZE>50MB DEVTYPE=DASD	For large data sets on all other devices, the blocksize is set for half- or third-track blocking.
SET	BLKSIZE=11476 INC DEVTYPE=3380	For all other data sets on 3380 devices, the blocksize is set to quarter-track blocking.
SET	BLKSIZE=13682 INC DEVTYPE=3390	For all other data sets on 3390 devices, the blocksize is set to quarter-track blocking.
SET	BLKSIZE=25600 INC DEVTYPE=TAPE	The blocksize for tape data sets is set to 25KB.
SET	BLKSIZE=27999 FORCE=YES INC PROGRAM=PRO031/	For all programs that start with PRO031, set the blocksize to 27999 and override any program blocksize.

SETEXPDT—Set Expiration Date for Data Set

The SETEXPDT function sets expiration dates for data sets being created.

The OS/390 expiration date for a data set prevents deletion of that data set by normal processing; that is, by disposition processing specified in a JCL DD statement, or by deletion through ISPF services. Proper use of the expiration date facility protects data sets from deletion while they are still useful; improper use of expiration dates can retain data sets long past their useful life, resulting in wasted DASD space and tape volumes.

SETEXPDT allows the data center to set standards for retention periods and to enforce those standards rigorously. Application of expiration dates can be based on data set names, on device types, on assigned pools, on size, or on any property allowed as a MAINVIEW SRM filter list parameter. In addition, the calculation of an expiration date can take into account the non-working days that occur in the retention period.

SETEXPDT also provides a convenient method of assigning symbolic expiration dates to selected data sets; for example, assigning CA-1 special-use dates (out-of-library, days since last used, and so on). New releases of CA-1 use ACCODE to determine special use tapes. The AC-CODE SET parameter supports this feature.

System Parameters

The system parameters that affect SETEXPDT are described in Table 3-6.

Table 3-6 System Parameters—SETEXPDT

Parameter	Purpose
DASDGENR	Specifies the only generic DASD names that are to be processed for specific functions if no space requirements are specified. Standard pooling functionality allows the JCL UNIT parameter to specify a pool name. If this is not desired, DASDGENR can be used to intercept data set allocations without space information. Such allocations will only be considered for subsequent processing if the unit is found in an internal table (3380, 3350, and so on) or if the unit is found in the DASDGENR list. DASDGENR affects functions DASDPOOL, DSNCHECK, SETEXPDT, and FORCECAT.
TAPEGENR	Specifies tape device generic names that some EasyPOOL functions will intercept. If you want to intercept all tape requests, specify ALLTAPE as the first generic name. (However, you cannot use ALLTAPE when JCLEXT=NO.) TAPEGENR affects functions DSNCHECK, FORCECAT, SETEXPDT, and TAPEPOOL.

Rule List Parameters

SET Statement (in member SMRLSTxx)

Parameter	Description
AC_CODE=xxxxxxx	Specifies the value in the IBM ACCODE field.
CAL=YES/NO	Adjusts calculated expiration date by the non-working days calendar
EVENTID=xxxxx	Identifies a user event defined in an SMEVNTxx member
EXPDT=yyddd or yyyddd	New expiration date in Julian format
REPLACE=YES/NO	Allows override to JCL-specified expiration date
RETPD=nnnn	Retention period in days

Rule List Parameter Explanations

AC_CODE=

Purpose: Specifies the value in the IBM ACCODE field. Normally, this field is used in conjunction with user-generated tape labels. CA's tape manager, CA-1, uses the field to indicate various special tapes, such as off-site, permanent hold, and so on.

Syntax: AC_CODE=xxxxxxx

Default: None

CAL=

Purpose: Specifies that the calculation of a new expiration date must consider non-working days in MAINVIEW SRM's non-working day calendar. If EXPDT is specified, CAL is ignored, since EXPDT directs a specific date. However, if RETPD is specified with CAL=YES, the new expiration date is calculated by adding the retention period and the number of non-working days in that period to the creation date.

Syntax: CAL=YES/NO

Default: CAL=YES

Note: ISPF function 3.4 does not correctly display expiration dates for JCL-defined VSAM data and index component expiration dates. LISTCAT does show the correct dates.

EVENTID=

Purpose: Specifies the identifier assigned to a user event in SMEVNTxx. This parameter will cause an event to be generated from this function.

Note: If EVENTID= is used on an FLST SET statement with MODE=INACT, the event will still be issued.

Syntax: EVENTID=xxxxx

where xxxxx is the 5-character string specified on the EVNTID parameter in SMEVNTxx.

Default: None

EXPDT=

Purpose: Specifies a data set expiration date. Both CAL and RETPD are ignored if EXPDT is specified.

The specified Julian date is not checked for leap year validity; that is, if you specify a non-leap-year with 366 days (for example, 97366), it is accepted.

Syntax: EXPDT=nnnnn

where nnnnn is a Julian date in the format yyddd to represent a date from 1900–1999.

OR

EXPDT=nnnnnnnn

where nnnnnnn is a Julian date in the format yyyyddd to represent a date from any year.

Default: None

REPLACE=

Purpose: Specifies whether an expiration date already specified for the data set (using JCL, for example) is overridden. If REPLACE=NO and an expiration date already exists for a data set, then the SETEXPDT function does not replace it. If REPLACE=YES, the SETEXPDT function sets the expiration date, regardless of any existing expiration date.

Syntax: REPLACE=YES/NO

Default: REPLACE=YES

RETPD=

Purpose: Specifies the retention period in days for a new data set. If CAL=YES is also specified, the retention period **and** any non-working days within that period are added to the data set's creation date to calculate the new expiration date. If CAL=NO is specified (or defaulted), the retention period is added to the data set's creation date to calculate the new expiration date; non-working days are ignored.

Note that a specification of EXPDT in the same SET statement overrides any RETPD specification.

Syntax: RETPD=nnnn

where *nnnn* is a number in the range 0–9999.

Default: None

Parameters Not Supported

The following filter list parameters are not supported for SETEXPDT:

- GDGVER
- VALUE

Usage Notes

SETEXPDT provides services only for data sets being created; data sets that already exist are not changed by SETEXPDT. For VSAM data sets, SETEXPDT works for JCL-defined VSAM but not for IDCAMS-defined VSAM.

Specification of the expiration date parameter (EXPDT) directs the usage of a specific date and overrides any specification of retention period (RETPD) and non-working days (CAL). If EXPDT is omitted and RETPD specified, the new expiration date is calculated by adding the retention period (in days) to the data set's creation date. If CAL=YES is also specified, the non-working days in the retention period (specified in the SMCALSxx member) are also added to the creation date.

If neither EXPDT nor RETPD is specified, no expiration date is set by the function, even though a data set may satisfy all filter list and rule list requirements.

The REPLACE parameter allows SETEXPDT to be applied in a restrictive or a lenient manner. The REPLACE parameter specifies whether a previously-specified expiration date is overridden by SETEXPDT. REPLACE=YES forces the use of the expiration date established in SETEXPDT, regardless of any expiration date or retention period already set. REPLACE=NO, however, accepts the original expiration date specification (in JCL, for example); the expiration date calculated by SETEXPDT is applied only if there is no existing expiration date specification.

Note that the SETEXPDT function must be specified in the SMFUNCxx member to be available and must be set active to provide services.

Messages

SETEXPDT generates message SVM3453I when it provides an expiration date for a data set:

```
SVM3453I  job,step,dd,dsn EXPDT (yyddd) ASSIGNED
```

SETEXPDT generates message SVM3348I when it the ACCODE field for the current allocation has been set to *ac_code*.

```
SVM3348I  job,step,dd,dsn ACCODE SET TO ac_code
```

Examples

The following example shows you how to set an expiration date for all data sets created by a batch job in the pool PROD, based on a retention period of 180 days. It overrides any expiration date already specified for the data set. You must take non-working days into account.

SMFUNCxx member

```
SET  NAME=SETEXPDT ACTIVE=YES  
      MSG=I SMF=I  
      FLST=64 RLST=73  
      DESC='SET EXPDT FOR DASD & TAPE'
```

The function is defined and activated.

SMFLST64 member

```
SET  MODE=ACT  
      INC POOL=PROD
```

This filter list specification is active. It selects all data sets in pool PROD.

SMRLST73 member

SET RETPD=180 REPLACE=YES CAL=YES INC XMODE=JOB	For all batch jobs (XMODE=JOB), the retention period is 180 days, plus the number of non-working days in that 180-day period.
SET RETPD=10 REPLACE=YES INC RETPD>10 JOBCLASS=T	Allow a maximum retention period of 10 days for data sets created by test jobs.

The following example shows you, for all data sets in pool PROD, how to:

- set a specific expiration date for GL93 data sets whose names do not end with TEST, if created by a batch job and if an expiration date has not been specified
- set a 30-day retention period for those data sets whose names end with TEST
- require the use of this new expiration in place of any expiration date already specified for the data set

SMFUNCxx member

SET NAME=SETEXPDT ACTIVE=YES MSG=I SMF=I FLST=5A RLST=7A DESC='SET EXPDT FOR DASD & TAPE'	The function is defined and activated.
--	--

SMFLST5A member

SET MODE=ACT INC POOL=PROD	This filter list specification is active. It selects all data sets in pool PROD.
-------------------------------	--

SMRLST7A member

SET EXPDT=94365 REPLACE=NO INC XMODE=JOB DSN=GL93/ LLQ-≠TEST	For all data sets created by a batch job, with a name starting with GL93 and ending with something other than TEST, and which do not already have an expiration date, an expiration date of the last day in 1994 is specified.
SET RETPD=30 REPLACE=YES INC DSN=GL93*.**.TEST	For all data sets starting with GL93 and ending with TEST, a 30-day retention period is forced.

SMSACSTE—Test ACS Routines

The SMSACSTE function provides services that enhance the process of testing ACS routines:

- ACS variables are displayed, allowing easier debugging of ACS routines.
- The data set can be removed from DFSMS-management, allowing testing of ACS routines without generating numerous new DFSMS-managed data sets.

Rule List Parameters

SET Statement (in member SMRLSTxx)

Parameter	Description
EVENTID=xxxxx	Identifies a user event defined in an SMEVNTxx member
SMSMANAGED=YES/NO	Specifies whether the data set is to remain DFSMS-managed

Rule List Parameter Explanations

EVENTID=

Purpose: Specifies the identifier assigned to a user event in SMEVNTxx. This parameter will cause an event to be generated from this function.

Note: If EVENTID= is used on an FLST SET statement with MODE=INACT, the event will still be issued.

Syntax: EVENTID=xxxxx

where xxxxx is the 5-character string specified on the EVNTID parameter in SMEVNTxx.

Default: None

SMSMANAGED=

Purpose: Specifies whether the data set is to remain DFSMS-managed. NO removes the data set from DFSMS control; YES leaves the data set under DFSMS management.

Syntax: SMSMANAGED=YES/NO

Default: SMSMANAGED=YES

Parameters Not Supported

The following filter list parameters are not supported for SMSACSTE:

- GDGVER
- VALUE

Usage Notes

SMSACSTE applies only to data sets that are DFSMS-managed. The function displays most ACS variables and also allows the data set to be removed from DFSMS management. SMSACSTE is activated for data sets with a status disposition of NEW or MOD.

SMSACSTE recognizes a DFSMS-managed data set by the assignment of a storage class to the data set. If the data set is not DFSMS-managed, SMSACSTE does not process the data set. No messages are produced, the SMSMANAGED parameter is ignored and the trace facility shows no activity. Regardless of any construct assignments by SMSACSDC, SMSACSMC, or SMSACSSG, the SMSACSTE function does not process a data set unless it has a storage class assigned.

The SMSMANAGED parameter specifies whether or not the data set remains under DFSMS management. SMSMANAGED=NO clears the storage class for the data set and so forces the data set to be non-DFSMS-managed. SMSMANAGED=YES does not change the storage class; the data set keeps its original storage class assignment and remains DFSMS-managed.

Note: Setting SMSMANAGED to YES does not change a non-DFSMS-managed data set to DFSMS-managed because non-DFSMS-managed data sets are not processed by SMSACSTE.

Due to information that is available from OS/390 at the time this function is invoked, the only MAINVIEW SRM filter list parameters that have values are those that correspond to DFSMS ACS variables.

These parameters are:

DATACLAS	ENVIR	MGMTCLAS	SMSMANAGED
DD	EXPDT	NQUAL	STEPACCTn
DSN	HLQ	PGM	STOGROUP
DSNn	JOB	RACF	STORCLAS
DSNTYPE	JOBACCTn	RECORG	UNIT
DSORG	LLQ	RETPD	USER
DSTYPE	MAXSIZE	SIZE	VOL
			XMODE

ACS Variables

The ACS variables that are displayed by SMSACSTE shown in Table 3-7.

Table 3-7 ACS Variables Displayed by SMSACSTE

Variable	Explanation
ACCT_JOB ACCT_STEP ACSENVIR	First five job account fields First five step account fields ACS routine environment (RECALL, RECOVER, CONVERT, ALLOC, STORE, CHANGED, CTRANS, or other)
APPLIC DATA_CLASS DDN DSN DSNTYPE DSORG DSOWNER DSTYPE	Application (if RACF is installed) Data class DDname Data set name Data set name type Data set organization (PS, PO, VS, DA) Data set owner name (if RACF is installed) Data set type (GDS, PERM, TEMP). Note that the MAINVIEW SRM filter list parameter DSTYPE uses the value GDG rather than GDS, to be consistent with industry usage.
EXPDT GROUP JOB MANAGEMENT_CLASS MAXSIZE	Expiration date RACF group Job name Management class Size of primary extent plus all secondary extents
NQUAL NVOL PGM RECORG RETPD SIZE	Number of qualifiers in the data set name Number of volumes Program name Record organization (KS, ES, RR, LS) Retention period Size of primary extent or of primary + 1 secondary extent
STORAGE_CLASS STORAGE_GROUP UNIT USER VOL XMODE	Storage class Storage group Device type generic name User ID Volume serial number Execution mode (BATCH, TSO, STC)

Variables with a blank or null value are not displayed.

Messages

SMSACSTE issues message SVM0304I to display ACS variables:

SVM0304I *variable(value)*

where *variable* is the name of the ACS variable, and *value* is the value of that variable; for example:

```
SVM0304I  JOB(EMPCRMX) DDN(DD2) PGM(IEFBR14)
          DSN(EMPCRM.SMSACSTE.TEST) NQUAL(03)
SVM0304I  ACCT_JOB1(3500)
SVM0304I  SIZE(46KB) MAXSIZE(736KB) XMODE(JOB)
SVM0304I  DSNTYPE(NUL) DSTYPE(PERM) DSORG(PS)
SVM0304I  USER(EMPCRM) GROUP(BABEMP)
SVM0304I  ACSENVIR(ALLOC) UNIT(BABDA)
SVM0304I  STORAGE_CLASS(BBSC) STORAGE_GROUP(BBSG)
```

If SMSMANAGED=NO is specified, message SVM0307I SMSMANAGED=NO APPLIED is written to indicate that any storage class has been removed from the data set.

Note that the SMSACSTE function must be specified in the SMFUNCxx member to be available and must be set active to provide service.

This example shows you how to use SMSACSTE to assist in debugging ACS routines.

SMFUNC04 member

SET	NAME=SMSACSTE ACTIVE=YES	The function is defined and activated.
	MSG=I SMF=I	
	FLST=04 RLST=02	
	DESC='TEST ACS ROUTINES'	

SMFLST04 member

SET	MODE=ACT	Select test data sets.
	INC DSN=SYSPGMR.ACSTEST./	

SMRLST02 member

SET	SMSMANAGED=NO	If the third data set name qualifier begins with NOSMS, do not allow the data set to be DFSMS-managed.
	INC DSN3=NOSMS*	
SET	SMSMANAGED=YES	If the third data set name qualifier begins with SMS, let it stay DFSMS-managed.
	INC DSN3=SMS*	

SMSMCREN—Identify Data Set Rename Requiring Management Class Change

The SMSMCREN function identifies data set renames that require a change in management class; that is, the management class for the new name is different from the management class for the old name.

Management classes allow enforcement of storage retention policies at a data set level. Management classes can affect data set retention/expiration date, release of allocated but unused space, and DFHSM migration and backup. When a data set is allocated, the management class can be set explicitly by JCL, or by an ACS routine. This management class stays with the data set until it is deleted, or manually changed by the data center through DFSMS utilities. Renaming the data set can reclassify that data set into a different management class; however, DFSMS does not change the management class on a data set name change.

For data set renames that require a management class change, SMSMCREN detects and reports the management class for the original data set name and the management class for the new data set name.

Rule List Parameters

SET Statement (in member SMRLSTxx)

Parameter	Description
EVENTID=xxxxx	Identifies a user event defined in an SMEVNTxx member

Rule List Parameter Explanations

EVENTID=

Purpose: Specifies the identifier assigned to a user event in SMEVNTxx. This parameter will cause an event to be generated from this function.

Note: If EVENTID= is used on an FLST SET statement with MODE=INACT, the event will still be issued.

Syntax: EVENTID=xxxxx

where xxxxx is the 5-character string specified on the EVNTID parameter in SMEVNTxx.

Default: None

Parameters Not Supported

The following filter list parameters are not supported for SMSMCREN:

ALCTYPE	BLKSIZE	BUFSP
CISIZE	CONTIG	CURSPACE
DATACLAS	DD	DEVTYPE
DIR	DSNTYPE	ERASE
EXPDT	EXTENT	IMBED
LRECL	MAXSIZE	MGMTCLAS
MNTYPE	OWNER	POOL
PRISPACE	RECFM	RELEASE
REPL	RETPD	REUSE
ROUND	SECSPACE	SIZE
SMS	STOGROUP	STORCLAS
UNIT	VOL	VSAMCOMP

Usage Notes

SMSMCREN operates on any SMS-managed data set being renamed. If the management class changes for the renamed data set, SMSMCREN issues a MAINVIEW SRM message. If the management class does not change with a rename, SMSMCREN does not issue a message.

SMSMCREN does not change the management class.

SMSMCREN recognizes a change in management class by comparing the existing management class with the management class returned by a call to the DFSMS ACS routine and the MAINVIEW SRM SMSACSMC function. If the management class function returns the same management class that is already assigned to the data set, SMSMCREN does not report a change; if the returned management class is different, SMSMCREN does report a change. Note that if the SMSACSMC function's rule that applies to this data set has REPLACE=NO, no management class is returned to SMSMCREN, which correctly interprets this as no change in management class. Also, if the SMSACSMC function's rule assigns a null management class (MGMTCLAS=), no change is reported by SMSMCREN (because this is identical to the output of REPLACE=NO).

All filter list parameters are available for use in the SMSMCREN filter list; however, the SMSACSMC function is limited to certain parameters. The function description of SMSACSMC in the EasySMS section of this manual details the filter list parameters available to SMSACSMC. In addition to those parameters, when SMSACSMC is called by SMSMCREN, the ENVIR parameter will be set to RENAM.

With the exception of the EVENTID rule list parameter, all resource selection must be done through the FLST.

Note that the SMSMCREN function must be specified in the SMFUNCxx member to be available and must be set active to provide services.

Messages

The MSG parameter for this function, either in SMFUNCxx or in SMFLSTxx, should not be set to W, E, S, or N, since this would prevent the informational message provided by SMSMCREN from being produced.

SMSMCREN generates message SMS3399I when it identifies a data set rename that requires a management class change:

```
SVM3399I  job,step,dd,dsn MGMTCLAS SHOULD BE ALTERED FROM
          oldclass TO newclass
```

where *oldclass* identifies the original management class and *newclass* identifies the new management class that should be assigned

```
SVM3400I  job,step,dsn SMSMCREN LOCATE FAILED RC = return code
SVM3401I  job,step,dsn SMSMCREN ACS CALL FAILED RC = return code
```

Examples

The following example shows you how to identify all renames that require a change in management class.

SMFUNCxx member

```
SET      NAME=SMSMCREN ACTIVE=YES
          MSG=I SMF=I
          FLST=76
          DESC='INVOKE SMS ON RENAME'
```

The function is defined and activated. There is no RLST specification.

SMFLST76 member

```
SET      MODE=ACT
          INC DSN=/'
```

This filter list specification is active. Selected resources are all data sets.

The following example shows you how to identify data set renames that require a change in management class, except for data sets that have a second name qualifier beginning with DEV or are being renamed by a job running in class T.

SMFUNCxx member

SET	NAME=SMSMCREN	ACTIVE=YES	The function is defined and activated. There is no RLST specification.
	MSG=I	SMF=I	
	FLST=MC		
	DESC='INVOKE SMS ON RENAME'		

SMFLSTMC member

SET	MODE=ACT	This filter list specification is active. It selects data sets without a second name qualifier beginning with DEV or in jobs in class T.
	EXC DSN2=DEV/	
	EXC JOBCLASS=T	

SMSSELECT—Control Selection of Volumes from SMS Storage Groups

The SMSSELECT function allows control over the selection of volumes from a storage group for DFSMS-managed data sets.

Volumes may be selected from the DFSMS-assigned storage group based on several criteria. SMSSELECT gives the data center the capability of assigning a specific volume based on the same volume selection techniques that are available in DASDPOOL:

- Best fit to a single extent on all volumes in the storage group.
- Volume with the largest percentage of free space.
- Volume with the largest single contiguous extent.
- Selection by performance statistics collected by MAINVIEW SRM.

A VOLSEL parameter must be assigned.

DFSMS Subpools

SMS data sets can also be *subgrouped* to certain DFSMS volumes within a STORGRP. This is done by assigning SMSPOOLS to data sets with the parameters SMSPOOL and SMSPOOL_EXT. For these parameters to be processed, the system options SMS_ALLOC and SMS_EXTEND must be set to Yes. Please note that this functionality is limited to denying volumes outside of an DFSMS-defined pool. Volume selection cannot be directed outside of the data set's current STORGRP. If multiple pools are specified, the first pool with sufficient space for the data set is assigned. The first pool specified is the primary pool; all others are alternate pools. See Chapter 5, "How to Define DFSMS Subpools," for information on defining the SMSPOOLxx definition parameter.

Critical Data Set Isolation

EasyPOOL allows you to identify a list of critical data sets in your environment that should never be placed on the same volume as any other data set in the list. EasyPOOL may also use data gathered by the Space Collector about the EMC Symmetrix environments to ensure that critical data sets are not only on separate logical volumes but are placed on separate physical volumes to reduce data set contention.

Note: The MAINVIEW SRM Reporting component must be installed and running to access EMC Symmetrix data.

It is important to consider, however, that CRITDSN is resource intensive and should be used only for a *small* list of critical data sets. It should not be used without considering the system impact. See Chapter 4, “How to Define Critical Data Set Lists,” for information on defining the SMCRTxx definition parameter.

System Parameters

The system parameters that affect SMSSELC are described in Table 3-8.

Table 3-8 System Parameters—SMSSELC

Parameter	Purpose
CRITLIST	Specifies the suffix of an SMCRTxx parameter member.
HISTDAYS	Specifies the number of days of data to retrieve from the Performance Collector and temporarily store in a data space. Values are 0-14 (7 or 14 recommended), which indicate the number of days of data to be retrieved. The default value is 0, which indicates that the data space is created, but no historical performance data is gathered and no HISTDPO pooling can be performed. If this value is changed, the Performance Collector (SVSGP) must be stopped and restarted. For a valid HISTDPO test, do not specify a number of HISTDAYS higher than the total number of days collected in the Performance Collector.
SMS_ALLOC	Specifies to EasyPOOL that SMSSELC will be driven during DADSM ALLOCATE. If a POOL is coded in SMSPOOL, the current volume will be compared to the volumes in the POOL. If the current volume is not in a POOL assigned to the data set, the volume will be rejected with a DADSM return code of 4.
SMS_EXTEND	Specifies to EasyPOOL that SMSSELC will be driven during DADSM EXTENDNV (new volume). If a POOL is coded in SMSPOOL_EXT, the current volume will be compared to the volumes in the POOL. If the current volume is not in a POOL assigned to the data set, the volume will be rejected with a DADSM return code of 4.

Rule List Parameters

SET Statement (in member SMRLSTxx)

Parameter	Description
CRITBIAS=nnnn	interval for device selection based on performance
CRITEMC=YES/NO	specifies whether the volume meeting CRITDSN criteria includes EMC physical volumes
CRITFAIL=YES/NO	defines allocation if a volume meeting the criteria cannot be found
CRITLIST=xxxxxxx	name of the table containing allocation volumes
DPORDEF=nnn	default response time for device selection
DPORMAX=nnn	maximum response time target for device selection
DPORMIN=nnn	minimum response time target for device select
DPORSEP=nnn	separation factor for device selection

Parameter	Description
DPOWIND= <i>nnnn</i>	Interval for device selection based on performance
EVENTID= <i>xxxxx</i>	Identifies a user event defined in an SMEVNTxx member
HDPORDEF= <i>nnn</i>	Default response time for device selection
HDPORMAX= <i>nnn</i>	Maximum response time target for device selection
HDPORMIN= <i>nnn</i>	Minimum response time target for device selection
HDPORSEP= <i>nnn</i>	Separation factor for device selection
HDPOSTIM= <i>nnnn</i>	Starting time each day for a range of snapshots to be analyzed
HDPOETIM= <i>nnnn</i>	Ending time each day for a range of snapshots to be analyzed
HDPODAYS=(<i>MO,TU,WE,TH FR,SA,SU</i>)	Day(s) of the week used in determining performance statistics for pooling
RAIDDEVTYPE= <i>xxxxxxxxx</i>	Allows the user to require a specific RAID device type for an allocation (EMC/RDFEMC/MIRROREMC/PARITYEMC) (requires the MAINVIEW SRM Reporting component) For RAIDDEVTYPE to be honored, a VOLSEL routine <i>must be</i> assigned.
SMSPOOL=(<i>xxxxxxxxx,xxxx xxxx,...</i>)	Defines an SMSPOOL(s) to be used during DADSM ALLOCATE
SMSPOOL_EXT=(<i>xxxxxxxxx,xxxxxxxxx,...</i>)	Defines an SMSPOOL(s) to be used during DADSM EXTEND
VOLSEL= <i>xxxxxxxxx</i>	Volume selection criteria (BESTFIT, PERCENT, MAXSPACE, DPO, HISTDPO, CRITDSN)

Rule List Parameter Explanations

CRITBIAS=

Purpose: Defines the number of data sets that can reside on the volume before the current allocation.

Syntax: CRITBIAS=*xxxx*

where *xxxx* is a number 1–9999.

Default: None

Note: This parameter works only in conjunction with VOLSEL=CRITDSN.

CRITEMC=

Purpose: Specifies whether the volume meeting CRITDSN criteria includes EMC physical volumes. If yes, data sets specified with CRITDSN should not reside on the same EMC Physical volume nor the same MVS Logical volume. If no, data sets specified with CRITDSN may reside on the same EMC physical volume. Default is no.

Syntax: CRITEMC=*YES/NO*

Default: CRITEMC=NO

Note: This parameter works only in conjunction with VOLSEL=CRITDSN.

CRITFAIL=

Purpose: Defines the allocation process if a volume meeting the criteria cannot be found. If Yes, allocation fails; if No, allocation is made to the best available volume. Like USELIM, if an acceptable volume cannot be found in the first pool, the best volume from the first pool is saved, and the system tries the next pool(s). If no acceptable volume is found, CRITFAIL is processed on the saved volume.

Syntax: CRITFAIL=*YES/NO*

Default: CRITFAIL=NO

Note: This parameter works only in conjunction with VOLSEL=CRITDSN.

CRITLIST=

Purpose: The table that contains allocation volumes.

Syntax: CRITLIST=*xxxxxxxx*

Default: None

Note: This parameter works only in conjunction with VOLSEL=CRITDSN.

DPORDEF=

Purpose: Specifies the default response time for device selection based on DASD performance. The default value is used when response time information has not been accumulated for a device, such as when it has just been brought online.

DPORDEF is useful only when specified in conjunction with DPORMIN/MAX.

Syntax: DPORDEF=*nnn*

where *nnn* specifies the default response time in milliseconds that will be substituted for unavailable information for a specific device.

Default: DPORDEF=50

DPORMAX=

Purpose: Specifies the maximum response time objective for device selection based on DASD performance.

DPORMAX is normally specified in conjunction with DPORMIN.

Note that the DASD Performance Optimization feature requires that VOLSEL=DPO be specified in order to select a device based on performance.

Syntax: DPORMAX=*nnn*

where *nnn* specifies the maximum response time in milliseconds that will be considered for device selection based on performance.

Default: None

DPORMIN=

Purpose: Specifies the minimum response time objective for device selection based on DASD performance.

DPORMIN is normally specified in conjunction with DPORMAX.

Note that the DASD Performance Optimization feature requires that VOLSEL=DPO be specified in order to select a device based on performance.

Syntax: DPORMIN=*nnn*

where *nnn* specifies the minimum response time in milliseconds that will be considered for device selection based on performance.

Default: None

DPORSEP=

Purpose: Specifies the response time value that is used to force selection of different volumes for data sets in the same jobstep. When multiple data sets are allocated in a single jobstep and device selection based on DASD performance is specified for some or all, this separation factor is added to the current response time for previously used volumes in order to increase the likelihood that new allocations will go to a different volume.

DPORSEP can only be specified in conjunction with DPORMIN/MAX.

Note that the DASD Performance Optimization feature requires that VOLSEL=DPO be specified in order to select a device based on performance.

Syntax: DPORSEP=*nnn*

where *nnn* specifies the response time in milliseconds that will be used to enhance data set separation across volumes

Default: DPORSEP=10

DPOWIND=

Purpose: Specifies the window (length of the performance interval) that is used to analyze the response characteristics of DASD devices for selection based on performance.

Note that the DASD Performance Optimization feature requires that VOLSEL=DPO be specified in order to select a device based on performance.

Syntax: DPOWIND=*nnnn*

where *nnnn* specifies the number of seconds over which the performance analysis will be made.

Default: DPOWIND=1800 (this is the maximum data retained in the performance tables)

EVENTID=

Purpose: Specifies the identifier assigned to a user event in SMEVNTxx. This parameter will cause an event to be generated from this function.

Note: If EVENTID= is used on an FLST SET statement with MODE=INACT, the event will still be issued.

Syntax: EVENTID=*xxxxx*

where *xxxxx* is the 5-character string specified on the EVNTID parameter in SMEVNTxx.

Default: None

HDPODAYS=

Purpose: Specifies the day(s) of the week to be used in determining performance statistics for pooling. This parameter is available only when VOLSEL=HISTDPO.

Syntax: HDPODAYS=(*MO, TU, WE, TH, FR, SA, SU*)

Default: HDPODAYS=(*MO, TU, WE, TH, FR, SA, SU*)

HDPOETIM=

Purpose: Specifies the ending time (each day) for a range of snapshots to be analyzed. This allows analysis of data from the same time range, or shift, each day. This parameter is available only when VOLSEL=HISTDPO.

Syntax: HDPOETIM=*nnnn*

where *nnnn* specifies a time in 24-hour format.

Default: HDPOETIM=2359

HDPORDEF=

Purpose: Specifies the default response time for device selection based on DASD performance. The default value is used when response time information has not been accumulated for a device, such as when it has just been brought online.

HDPORDEF is useful only when specified in conjunction with HDPORMIN/MAX. This parameter is available only when VOLSEL=HISTDPO.

Syntax: HDPORDEF=*nnn*

where *nnn* specifies the default response time in milliseconds that will be substituted for unavailable information for a specific device.

Default: HDPORDEF=50

HDPORMAX=

Purpose: Specifies the maximum response time objective for device selection based on DASD performance.

HDPORMAX is normally specified in conjunction with HDPORMIN. This parameter is available only when VOLSEL=HISTDPO.

Syntax: HDPORMAX=*nnn*

where *nnn* specifies the maximum response time in milliseconds that will be considered for device selection based on performance

Default: None

HDPORMIN=

Purpose: Specifies the minimum response time objective for device selection based on DASD performance.

HDPORMIN is normally specified in conjunction with HDPORMAX. This parameter is available only when VOLSEL=HISTDPO.

Syntax: HDPORMIN=*nn*

where *nn* specifies the minimum response time in milliseconds that will be considered for device selection based on performance.

Default: None

HDPORSEP=

Purpose: Specifies the response time value that is used to force selection of different volumes for data sets in the same jobstep. When multiple data sets are allocated in a single jobstep and device selection based on DASD performance is specified for some or all, this separation factor is added to the current response time for previously used volumes in order to increase the likelihood that new allocations will go to a different volume. This parameter is available only when VOLSEL=HISTDPO.

Syntax: HDPORSEP=*nn*

where *nn* specifies the response time in milliseconds that will be used to enhance data set separation across volumes.

Default: HDPORSEP=10

RAIDDEVTYPE=

Purpose: Allows the user to require a specific RAID device type for an allocation. If a particular hardware type is chosen, the candidate list of volumes will be limited to devices that have the requested attribute flag enabled. This parameter requires the MAINVIEW SRM Reporting component to be installed and running.

Note: For this feature to be used under SMSSELC, a data set must be assigned a VOLSEL parameter.

Syntax: RAIDDEVTYPE=xxxxxxxx

where *xxxxxxxx* is one of the following:

EMC	device must be an EMC RAID device
RDFEMC	device must be EMC and have the RDF flag enabled
MIRROREMC	device must be EMC and have the MIRROR flag enabled
PARITYEMC	device must be EMC and have a parity flag enabled

Default: None

SMSPOOL=

Purpose: Specifies 1 to 15 SMSPOOL(s) that can be used to limit volumes to be used in a primary allocation (the first volume) during DADSM ALLOCATE.

Syntax: SMSPOOL=(*xxxxxxxx,xxxxxxxx,...*)

where *xxxxxxxx* is an SMSPOOL.

Default: None

Note: The SMSPOOL(s) must contain a subset of volumes from the SMS STORGRP assigned to the data set, or the allocation will fail. If multiple pools are specified, the first pool with sufficient space for the data set is assigned. The first pool specified is the primary pool; all others are alternate pools. In addition, SMS_ALLOC must be set to *Yes* under System parameters.

SMSPOOL_EXT=

Purpose: Specifies 1 to 15 SMSPOOL(s) that are to be used to limit volume selection during DADSM EXTENDNV on an extend to a new volume (anything after the first volume). Any SMS volumes can be coded in the pools. However, allocation to volumes will only be directed to volumes in the storage group assigned to the current data set.

Syntax: SMSPOOL_EXT=(*xxxxxxxx,xxxxxxxx,...*)

where *xxxxxxxx* is an SMSPOOL.

Default: None

Note: The SMSPOOL(s) must contain a subset of volumes from the SMS STORGRP assigned to the data set, or the allocation will fail. If multiple pools are specified, the first pool with sufficient space for the data set is assigned. The first pool specified is the primary pool; all others are alternate pools. In addition, SMS_EXTEND must be set to *Yes* under System parameters

VOLSEL=

Purpose: Specifies that volume selection within a pool should be based on current volume space availability.

Syntax: *VOLSEL=BESTFIT/CRITDSN/DPO/HISTDPO/MAXSPACE/PERCENT*

BESTFIT – specifies that the volume with the smallest contiguous extent that satisfies the primary allocation should be selected

CRITDSN – controls allocations by specifying data sets that should not reside on the same volume

Warning! CRITDSN is resource intensive and should be used only for a *small* list of critical data sets. It should not be used without considering the system impact.

DPO – specifies volume selection based on performance statistics accumulated by MAINVIEW SRM

HISTDPO – specifies volume selection based on historical volume performance

Note: HISTDPO requires more resources than normal allocations.

MAXSPACE – specifies that the volume with the largest single contiguous extent should be selected

PERCENT – specifies that the volume with the largest amount of free space should be selected

Default: None

Parameters Not Supported

The following filter list parameters are not supported for SMSSELCCT:

- GDGVER
- VALUE

Usage Notes

SMSSELCCT provides services for DFSMS-managed data sets, both VSAM and non-VSAM, with a status of NEW or MOD.

SMSSELCCT will filter only those volumes placed on the DFSMS Primary List that meet *all* the Storage Class requirements. Refer to "DFSMS/MVS VnRn DFSMSdfp Storage Administration Reference" for further information on why volumes are not placed on the Primary List.

SMSSELCCT does not require or use any information from the pool definitions in the SMSPOLxx parmlib member or the DASDPOOL filter/rule lists.

A VOLSEL parameter must be assigned.

Volume Selection

Volumes are selected from the storage group assigned to the DFSMS-managed data set. The VOLSEL rule list parameter specifies the method of volume selection. The selection techniques are described in Table 3-9.

Table 3-9 Volume Selection Techniques

Parameter	Explanation
BESTFIT	The volume with the smallest contiguous extent that satisfies the primary allocation amount is selected
PERCENT	The volume with the largest percentage of free space is selected
MAXSPACE	The volume with the largest single contiguous extent is selected
DPO	A volume is selected based on the performance statistics accumulated by the MAINVIEW SRM DASD Performance Optimization facility and the SMSSELCCT rule list parameters DPORDEF, DPORMAX, DPORMIN, DPORSEP, and DPOWIND
CRITDSN	Controls allocations by specifying data sets that should not reside on the same volume. CRITDSN is resource intensive and should be used only for a <i>small</i> list of critical data sets. It should not be used without considering the system impact.

Messages

SMSSELECT issues message SVM3460I when a data set is assigned to a volume in a storage group:

SVM3460I *job,step,dd,dsn* VOL (*volser*) SELECTED

SMSSELECT issues one or more of the following messages:

SVM3353I *job,step,dd,dsn* CRITLIST = *crit_list*, CRITBIAS = *crit_bias*,
CRITFAIL = *crit_fail*, CRITEMC = *crit_emc*

SVM3354I *job,step,dd,dsn* CRITLIST FAILED ALL ELIGIBLE VOLUMES.
ALLOCATION REJECTED

SVM3360W *job,step,dd,dsn* ERROR ON CATALOG CALL. R15 = *rc*

SVM3361W *job,step,dd,dsn* ERROR ON RAID API CALL. R15 = *rc*

DASD Performance Optimization

The MAINVIEW SRM DPO facility is a specialized algorithm used during data set allocation to select a volume based on short-term historical performance data.

During I/O, the amount of time needed to complete the read or write operation is called the *response time*. The response time for a device can vary greatly due to the number of tasks competing for access to the volume and the channel path(s) to the volume. The DPO facility records device response times every 30 seconds and tables this performance data, maintaining 30 minutes worth of data in storage.

During allocation, the DPO facility scans the response time tables for the pooled devices over the indicated time interval; for example, for the last two minutes. The volume selected for allocation is the volume with the best response time. Alternatively, DPO can be used to select volumes based on response time objectives; for example, to select a volume that can provide a response time in the range of 20 to 40 milliseconds for some data sets or jobs and a volume that can provide a response time in the range of 40 to 60 milliseconds for other less response-critical data sets or jobs.

Within the MAINVIEW SRM ISPF interface, performance data can be displayed for each pool.

Starting and Stopping DPO

The DPO facility is started and stopped automatically as part of the SVOS started task.

Resources

DPO maintains a table of performance data above the 16MB line in extended private storage. The table contains one entry for each DASD UCB for 30 minutes (sampled twice a minute); each entry is 16 bytes long.

Usage

VOLSEL=DPO must be specified in the SMSSELECT rule for all data sets that are to use the DASD Performance Optimization allocation method.

The DPOWIND rule list parameter is used to specify the time interval over which the performance analysis is performed. For example, DPOWIND=120 indicates that the device with the best response time in the last 120 seconds should be selected for allocation. Since 30 minutes worth of data is retained, 1800 seconds is the logical upper limit for the DPOWIND operand.

Allocation based on response time objectives, rather than just the best response time, is specified in the DPORMIN and DPORMAX rule list parameters. These parameters set the minimum and maximum response times that are desired for the device to receive the allocation. For example, DPORMIN=20 and DPORMAX=40 will attempt to select a device that can provide a response time in the range of 20 to 40 milliseconds. If both DPORMIN and DPORMAX are zero, the best performing device is selected.

When multiple data sets are allocated in the same jobstep and some or all are using DPO allocation, the DPORSEP parameter is used to prevent all data sets from allocating to the same device, thus possibly degrading its performance significantly. The separation factor in DPORSEP is added to a previously used device's response time and the resulting figure is used in the selection process. This will artificially increase the device's response time, and cause subsequent allocations in the same job step to be allocated to another device with the desired response time characteristics.

In pools with fairly even performance statistics, the value of DPORSEP should generally be the average difference in response times on the different volumes. In pools with large differences in performance, such as pools that contain both cached and non-cached devices, allocations will concentrate on the fastest volumes, regardless of the separation factor.

In order to get a valid test of the DPO options, you should wait 30 minutes after you start DPO before you try to pool using DPO. It will take 30 minutes to collect the a full set of performance data. Any DPO-based allocations made before a full set of performance data has been collected may not reflect your actual selection algorithm.

Performance Data Display

The MAINVIEW SRM interface displays performance information for the devices in a pool. From the MPOOL view, type the **P** action line command to display performance data. The MVOLPER view (Figure 3-1) displays performance data (over the last 5 minute interval, maintained by DPO) for a selected pool, SMS pool, EMC physical device, or IBM physical device.

Figure 3-3 MVOLPER View

```

02JAN2001 13:29:57 ----- INFORMATION DISPLAY -----
COMMAND ==>
CURR WIN ==> 1          ALT WIN ==>
>W1 =MVOLPER=====SJSG=====02JAN2001==13:29:57====MVSRM=====16
C Volume UCB      Resp Queue  Pend  Disc  Conn  Busy  Serv  EXCPs # of SSI
- - - - - Addr    Time   Time   Time  Time  Time  Time  Time  ----- UCBs Targ
SMS011 85A3      4.43   0.00   0.17   0.00   4.26   0.03   4.26     3    1 SJSG
SMS301 853E      4.43   0.00   0.21   0.00   4.22   0.03   4.22     3    1 SJSG
SMS002 83BD      4.42   0.00   0.12   0.08   4.22   0.04   4.30     3    1 SJSG
SMS303 8540      4.39   0.00   0.21   0.00   4.18   0.03   4.18     3    1 SJSG
SMS001 8355      4.38   0.00   0.12   0.08   4.18   0.03   4.26     3    1 SJSG
SMS004 8357      0.62   0.00   0.25   0.12   0.25   0.00   0.37     1    1 SJSG
SMS21B 8524      0.50   0.00   0.25   0.00   0.25   0.00   0.25     1    1 SJSG
SMS306 8543      0.50   0.00   0.25   0.00   0.25   0.00   0.25     1    1 SJSG
SMS305 8542      0.50   0.00   0.25   0.00   0.25   0.00   0.25     1    1 SJSG
SMS304 8541      0.50   0.00   0.25   0.00   0.25   0.00   0.25     1    1 SJSG
SMS25A 8358      0.50   0.00   0.25   0.00   0.25   0.00   0.25     1    1 SJSG
SMS003 8356      0.50   0.00   0.25   0.00   0.25   0.00   0.25     1    1 SJSG
SMS302 853F      0.49   0.00   0.12   0.12   0.25   0.00   0.37     1    1 SJSG
MSG00 8576      0.37   0.00   0.12   0.00   0.25   0.00   0.25     1    1 SJSG
SMSA00 85A2      0.00   0.00   0.00   0.00   0.00   0.00   0.00     0    1 SJSG
SMSA01 8579      0.00   0.00   0.00   0.00   0.00   0.00   0.00     0    1 SJSG

```

Historical DASD Performance Optimization

The MAINVIEW SRM HISTDPO facility allows you to pool data sets based on performance data gathered over a period of up to two weeks. Historical performance data is obtained using Performance Collector in the MAINVIEW SRM Reporting component, which must be licensed and operational at your site.

Data is stored in a data space to decrease the access time required to obtain the performance data during allocation. The data is averaged and compared against the options you choose to determine the optimum volume.

Usage

The system parameter, HISTDAYS, specifies the number of days of data to retrieve from the Performance Collector and temporarily store in a data space. Values are 0-14 (7 or 14 recommended), which indicate the number of days of data to be retrieved. The default value is 0, which indicates that the data space is created, but no historical performance data is gathered and no HISTDPO pooling can be performed. If this value is changed, the Performance Collector (SVSGP) must be stopped and restarted. For a valid HISTDPO test, do not specify a number of HISTDAYS higher than the total number of days collected in the Performance Collector.

VOLSEL=HISTDPO must be specified in the TAPEPOOL rule for all data sets that are to use the HISTDPO allocation method.

Starting time, ending time, and days of collection are specified in HDPOSTIM, HDPOETIM, and HDPODAYS respectively.

Allocation based on response time objectives, rather than just the best response time, is specified in the HDPORMIN and HDPORMAX rule list parameters. These parameters set the minimum and maximum response times that are desired for the device to receive the allocation. For example, HDPORMIN=20 and HDPORMAX=40 will attempt to select a device that can provide a response time in the range of 20 to 40 milliseconds. If both HDPORMIN and HDPORMAX are zero, the best performing device is selected.

When multiple data sets are allocated in the same jobstep and some or all are using HISTDPO allocation, the HDPORSEP parameter is used to prevent all data sets from allocating to the same device, thus possibly degrading its performance significantly. The separation factor in HDPORSEP is added to a previously used device's response time and the resulting figure is used in the selection process. This will artificially increase the device's response time, and cause subsequent allocations in the same job step to be allocated to another device with the desired response time characteristics.

In pools with fairly even performance statistics, the value of HDPORSEP should generally be the average difference in response times on the different volumes. In pools with large differences in performance, such as pools that contain both cached and non-cached devices, allocations will concentrate on the fastest volumes, regardless of the separation factor.

Examples

The following example shows you how to assign SMS volumes.

SMFUNC22 member

```
SET  NAME=SMSSELECT ACTIVE=YES
      MSG=I SMF=I
      FLST=53 RLST=26
      DESC='ASSIGN VOL FROM STOGROUP'
```

SMFLST53 member

```
SET  MODE=ACT                               Select all DFSMS data sets.
      INC STOGROUP=/
```

SMRLST26 member

```
SET  VOLSEL=BESTFIT                         Select the volume with the
      INC STOGROUP=APPLA                     extent that is closest to the
                                              data set's primary extent
                                              request for all data sets in
                                              storage group APPLA.

SET  VOLSEL=MAXSPACE                       Select the volume with the
      INC DSN=DIV325.PAY*.*.MSTR             largest single extent for payroll
                                              master files in division 325.

SET  VOLSEL=PERCENT                         Select the volume with the
      INC JOBCLASS=T                         largest percentage of free
                                              space for all DFSMS-managed
                                              data sets in jobs running in
                                              class T.
```

The following example shows you how to require all initial allocations (first volume) for an SMS-managed data set with a second qualifier of PARMS to allocate on SMS-managed volumes SMS001 to SMS009. This example requires that all extends to a new volume for SMS-managed data sets with a second qualifier of PARMS go to SMS-managed volumes SMS010 and SMS011.

Note: If the SMS-managed pools that are used in the SMSSELECT parms of SMSPOOL and SMSPOOL_EXT do not include volumes in the data sets STORGRP, the allocation will fail.

SMSYS00 member

```
SMS_ALLOC=YES                               Process SMSSELECT
                                              during DASDM
                                              ALLOCATE processing.

SMS_EXTEND=YES                             Process SMSSELECT
                                              during DADSM
                                              EXTENDNV processing.
```

SMFUNC22 member

SET	NAME=SMSSELECT ACTIVE=YES MSG=I SMF=Y FLST=53RLST=26 DESC='ASSIGN VOL FROM STORGRP'	Define and activate the function.
SMSPOLxx member		
SET	NAME=SMSP1 INC VOL=SMS00/	Define SMSPOL named SMSP1 that includes all volumes that start with SMS00 (this will include SMS001 through SMS009).
SET	NAME=SMSP2 INC VOL=(SMS010,SMS011)	Define SMSPOL named SMSP2 that includes all volumes SMS010 and SMS011).
SMFLST53 member		
SET	MODE=ACT INC QUAL2=PARMS	Select all data sets that have PARMS as the second qualifier.
SET	MODE=INACT INC DADSM_FUNC=(ALLOCATE,EXTENDNV)	Do not process SMSPOL(s) for any other data sets. However, allow all other SMSSELECT options to be processed.
SMRLST26 member		
SET	SMSPPOOL=SMSP1,SMSPPOOL_EXT=SMSP2 INC QUAL2=PARMS	Allow the initial allocation only to the volumes defined in SMSP1, and allow extends to a new volume only to the volumes defined in SMSP2 only.

SPACCONV—Change Allocation Units to Blocks

The SPACCONV function converts space allocation values on new allocations from cylinders or tracks to blocks, for both primary and secondary allocation quantities.

Space allocation in blocks is advantageous because allocation in blocks is device independent; the system determines the number of tracks or cylinders necessary to hold the requested number of blocks, taking into account the device type. Allocation in units of cylinder or track is device-specific, because the size of the allocation unit varies with the device type.

SPACCONV is useful in allocating the correct amount of space when new device types are put into use, such as when converting from 3380 to 3390 devices. The old device type (for which the JCL space parameters are coded) is identified by the TRKLEN and TRKCYL parameters. SPACCONV automatically converts track and cylinder space requests into blocks, which, when applied by OS/390 during allocation to the new device type, allocates a proportionally larger or smaller total space.

System Parameters

The system parameters that affect SPACCONV are described in Table 3-10.

Table 3-10 **System Parameters—SPACCONV**

Parameter	Purpose
TRKCYL	Specifies the number of tracks per cylinder for the default device type. The value specified for 3380/3390/9345 devices should be fifteen. (Note that this specification is the same as the SCDS base configuration DEFINE under ISMF for DFSMS.) TRKCYL and TRKLEN are used by the DASDPOOL function to convert allocations in tracks or cylinders to megabytes for volume selection based on available space; for example, VOLSEL=BESTFIT. The information specified on these two parameters should reflect the devices that are most prevalent in your environment.
TRKLEN	Specifies the number of bytes per track for the default device type. Valid values are: 3380 – 47,476 3390 – 56,664 9345 – 46,456 Note that this specification is the same as the SCDS base configuration DEFINE under ISMF for DFSMS.) TRKCYL and TRKLEN are used by the DASDPOOL function to convert allocations in tracks or cylinders to megabytes for volume selection based on available space; for example, VOLSEL=BESTFIT. The information specified on these two parameters should reflect the devices that are most prevalent in your environment.

Rule List Parameters

SET Statement (in member SMRLSTxx)

Parameter	Description
EVENTID=xxxxx	Identifies a user event defined in an SMEVNTxx member
ROUND=YES/NO	Round allocation in blocks to cylinder boundary
TRKCYL=nn	Specifies the tracks per cylinder
TRKLEN=nnnnn	Specifies the bytes per track

Rule List Parameter Explanations

EVENTID=

Purpose: Specifies the identifier assigned to a user event in SMEVNTxx. This parameter will cause an event to be generated from this function.

Note: If EVENTID= is used on an FLST SET statement with MODE=INACT, the event will still be issued.

Syntax: EVENTID=xxxxx

where xxxxx is the 5-character string specified on the EVNTID parameter in SMEVNTxx.

Default: None

ROUND=

Purpose: Specifies that the converted space allocation should be rounded up to a cylinder boundary.

This is an optional parameter. If not used, the SMRLSTxx member can be omitted for this function.

Syntax: ROUND=YES/NO

Default: None

TRKCYL=

Purpose: Specifies the number of tracks per cylinder of the **source volume** for the SPACCONV function. The value is used to calculate the proper size of an allocation on new devices for specifications based on devices no longer used.

Syntax: TRKCYL=*nn*

where *nn* is a number in the range of 1–99.

Default: None (if not specified, the TRKCYL value in SMMSYSxx is used)

TRKLEN=

Purpose: Specifies the number of bytes per track of the **source volume** for the SPACCONV function. The value is used to calculate the proper size of an allocation on new devices for specifications based on devices no longer used.

Syntax: TRKLEN=*nnnnn*

where *nnnnn* is a number in the range of 1–99999.

Default: None (if not specified, the TRKLEN value in SMMSYSxx is used)

Parameters Not Supported

The following filter list parameters are not supported for SPACCONV:

- GDGVER
- VALUE

Usage Notes

SPACCONV applies to new DASD data sets (non-VSAM and JCL-defined VSAM). It does not apply to tape data sets or ISAM data sets. The original space allocation must be in cylinders or tracks; if it is already in blocks, no action is taken. Both primary and secondary allocation quantities are converted.

Conversion of space allocation from cylinders or tracks to blocks is based on the bytes/track and tracks/cylinder values specified in the TRKLEN and TRKCYL rule list parameters. The average block size is set to 9000 (this yields 5 blocks per track on both 3380 and 3390 devices). The calculated number of blocks is always increased by one. If ROUND is specified, rounding and alignment to a cylinder boundary is performed by OS/390. (Although an average block size of 9000 is used to calculate space, block size specified for the data set is not modified.)

If TRKLEN and/or TRKCYL are not specified in the rule list, the values are taken from the TRKLEN and TRKCYL parameters in the SMMSYSxx parmlib member. However, these SMMSYSxx parameters are used to represent the *normal* devices for allocation; accepting these values as a default causes the SPACCONV function to allocate the full size specified in the JCL.

The TRKLEN and TRKCYL specifications can have different values on different rules. This is useful when upgrading from two obsolete device types to a new device type, such as from 3350s and 3380s to 3390s.

Note: The ISPF data set information display in functions 3.4 and 3.2 do not always give the correct values for secondary space in blocks, particularly when the data set is allocated without a blocksize. The information provided by IEHLIST is correct, however.

Messages

The messages issued by SPACCONV are:

SVM3448I *job,step,dd,dsn* SPACE CONVERTED TO *prim/sec* BLOCKS

SVM3318I *job,step,dd,dsn* ROUND=y/n ASSIGNED

Note that the SPACCONV function must be specified in the SMFUNCxx member to be available and must be set active to provide services.

Example

This example shows you how to specify space conversion for unit specifications of 3350 and 3380.

SMFUNC33 member

```
SET  NAME=SPACCONV ACTIVE=YES  The function is defined and
    MSG=I SMF=I               activated.
    FLST=33 RLST=33
    DESC='CONV SPACE TO BLOCKS'
```

SMFLST33 member

SET MODE=ACT
INC UNIT=3350
INC UNIT=3380

The function is to run in active mode for all allocation requests to devices no longer used (3350s and 3380s).

SMRLST33 member

SET ROUND=YES
TRKLEN=47476 TRKCYL=15
INC UNIT=3380

X Convert space for allocations coded for 3380 devices. Round allocations to an integral number of cylinders.

SET TRKLEN=19069 TRKCYL=30
INC UNIT=3350

Convert space for allocations coded for 3350 devices.

SPACLIMI—Limit the Size of Space Allocation Requests

The SPACLIMI function limits the size of **new** data sets.

Setting and enforcing maximum sizes for data set space allocation is not directly possible in OS/390. Setting data set size limits is desirable because:

- Estimates coded in JCL SPACE statements may or may not be accurate.
- Excessive sizes for data sets which exist only for the life of a single job may never be detected, yet can cause other allocations to fail for lack of space.
- Unused space in permanent data sets may never be released (although the JCL parameter RLSE can be specified, there is no way to enforce this).

Excessive space allocations, both temporary and permanent, can unnecessarily consume significant DASD space, causing jobs to fail for lack of space and requiring the purchase of extra DASD devices.

The SPACLIMI function allows limits to be set on the size of the primary allocation for new data sets. Combined with the filter list parameter MAXSIZE, limits can also be placed on data sets based on their maximum possible size (primary plus the maximum secondary extents). Running the function in the simulate mode allows warnings to be given to users before tightly enforcing the limits.

System Parameters

The system parameters that affect SPACLIMI are described in Table 3-11.

Table 3-11 **System Parameters—SPACLIMI**

Parameter	Purpose
REJECT	Controls termination of processing at the first or last data set rejected by REJECT=YES in DASDPOOL or DSNCHECK. If termination is to take place on the first rejected data set, code REJECT=FIRST. If all data sets are to be processed before control is returned to OS/390 allocation, code REJECT=LAST.

Rule List Parameters

SET Statement (in member SMRLSTxx)

Parameter	Description
EVENTID=xxxxx	Identifies a user event defined in an SMEVNTxx member
LIMIT=nnnnnnnnKB/MB	Data set size limit in kilobytes or megabytes

Rule List Parameter Explanations

EVENTID=

Purpose: Specifies the identifier assigned to a user event in SMEVNTxx. This parameter will cause an event to be generated from this function.

Note: If EVENTID= is used on an FLST SET statement with MODE=INACT, the event will still be issued.

Syntax: EVENTID=xxxxx

where xxxxx is the 5-character string specified on the EVNTID parameter in SMEVNTxx.

Default: None

LIMIT=

Purpose: Specifies maximum allowable size for a data set primary allocation. This is a required parameter.

Syntax: LIMIT=nnnnnnnnKB/MB

where nnnnnnnn specifies the maximum allowed size of the new data set.

Default: None

Parameters Not Supported

The following filter list parameters are not supported for SPACLIMI:

- GDGVER
- VALUE

Usage Notes

SPACLIMI applies to *new* DASD data sets (VSAM and non-VSAM). If the requested amount of space for the data set being allocated is greater than the limit specified in the SPACLIMI rule, MAINVIEW SRM causes the job to fail with a JCL error. (A VSAM IDCAMS DEFINE will fail with a VSAM catalog return code of 68 – unable to allocate space on user volume.)

The limit is on the size of the primary allocation only. At allocation time, the requested primary allocation is converted to kilobytes using either the bytes/track and tracks/cylinder specifications in SMMSYSxx (TRKLEN and TRKCYL) or the average block length in the JCL SPACE parameter.

The LIMIT parameter must be specified; SPACLIMI takes no action without a rule specifying the limit for selected resources.

Note that the SPACLIMI function must be specified in the SMFUNCxx member to be available and must be set active to provide services.

The SPACRLSE function provides related space management functionality for new data sets.

Messages

The message issued by SPACLIMI when a limit is enforced is:

```
SVM3352I  job,step,dd,dsn REQUESTED SPACE nnnnKB/MB EXCEEDS LIMIT
```

where *nnnn* is the requested size in kilobytes or megabytes.

Examples

The following example shows you how to run SPACLIMI in simulate mode for production jobs and in active mode for test jobs and write SMF records only for production jobs.

SMFUNC1A member

```
SET  NAME=SPACLIMI  ACTIVE=YES  
      MSG=I  SMF=N  
      FLST=A1  RLST=A2  
      DESC='LIMIT SPACE ALLOCATION'
```

The function is defined and activated. The default for SMF records is none.

SMFLSTA1 member

SET	MODE=SIM SMF=I MSG=N INC JOBCLASS=P	The function is to run in simulate mode for production jobs. All messages for SPACLIMI processing on production jobs are to be written to SMF for later analysis, but no messages are to be written to the production job logs.
SET	MODE=ACT INC JOBCLASS=T	The function is to run in active mode for test jobs, causing jobs to fail if a data set exceeds the limit.
SMRLSTA2 member		
SET	LIMIT=50MB INC DSN=	Set the data set size limit for all data sets to 50 megabytes.

The following example shows you how to:

- impose size limits on all data sets except those in pool OVERSZ and those in started tasks

The size limit for TSO data sets is 1 MB and 75 MB for batch jobs data sets not in STD0 pools.

- restrict primary allocations for data sets in pools STD001-STD013 to 10% of their maximum size, with an overall limit of 100 MB

SMFUNC20 member		
SET	NAME=SPACLIMI ACTIVE=YES MSG=I SMF=I FLST=39 RLST=3A DESC='ENFORCE SIZE LIMITS'	The function is defined and activated.
SMFLST39 member		
SET	MODE=ACT EXC POOL=OVERSZ EXC XMODE=STC INC DSN=	The function is to run in active mode. Data sets being allocated in pool OVERSZ or data sets being allocated from started tasks are exempted from size limits.
SMRLST3A member		
SET	LIMIT=1MB INC XMODE=TSO	For all TSO allocations, limit data sets to a 1 MB primary extent.
SET	LIMIT=100MB INC MAXSIZE>1000MB POOL=STD0*	For STD0 pools, limit data sets potentially larger than 1000 MB to a 100 MB primary extent.
SET	LIMIT=10MB INC MAXSIZE>100MB POOL=STD0*	For STD0 pools, limit data sets potentially larger than 100 MB to a 10 MB primary extent.

SET	LIMIT=1MB INC MAXSIZE>10MB POOL=STD0*	For STD0 pools, limit data sets potentially larger than 10 MB to a 1 MB primary extent.
SET	LIMIT=100KB INC MAXSIZE>1MB POOL=STD0*	For STD0 pools, limit data sets potentially larger than 1 MB to a 100 KB primary extent.
SET	LIMIT=50KB INC DSN=/ POOL=STD0*	For STD0 pools, limit data sets with a maximum size less than 1 MB to a 50 KB primary extent.
SET	LIMIT=75MB INC XMODE=JOB	For all batch job allocations except those in the STD0 pools, limit data sets to 75 MB.

SPACRLSE—Release Data Set Space at Close

The SPACRLSE function causes unused space in physical-sequential data sets to be released when the data set is closed.

Inaccurate space estimates can allocate significantly more space in data sets than is actually needed. Excessive space requests are frequently made when the expected data set size is not known, particularly to avoid abending with a space related problem. While OS/390 provides a facility to release unused space from a new data set, it must be requested using the RLSE subparameter of the SPACE parameter in JCL. If RLSE is not specified in the JCL, the extra space in the data set is not freed until the data set is eventually deleted, or until the storage administrator manually searches for and releases that space.

Excessive space allocations unnecessarily consume significant DASD space, causing other jobs to fail for lack of space and requiring the purchase of extra DASD devices.

The SPACRLSE function allows automated release of unused space, regardless of JCL specifications.

Rule List Parameters

SET Statement (in member SMRLSTxx)

Parameter	Description
EVENTID=xxxxx	Identifies a user event defined in an SMEVNTxx member
RLSE=ALL/SEC/NO	Specifies whether space is released for <i>all</i> selected data sets, for only those selected data sets that have <i>secondary</i> allocations, or <i>no</i> data sets.

Rule List Parameter Explanations

EVENTID=

Purpose: Specifies the identifier assigned to a user event in SMEVNTxx. This parameter will cause an event to be generated from this function.

Note: If EVENTID= is used on an FLST SET statement with MODE=INACT, the event will still be issued.

Syntax: EVENTID=xxxxx

where *xxxxx* is the 5-character string specified on the EVNTID parameter in SMEVNT*xx*.

Default: None

RLSE=

Purpose: Specifies whether space is released for *all* selected data sets, for only those selected data sets that have *secondary* allocations, or *no* data sets. This is a required parameter.

Syntax: RLSE=*ALL/SEC/NO*

ALL	releases space for all data sets
SEC	releases space for data sets with a secondary allocation
NO	turns the release flag off

Default: None

Parameters Not Supported

The following filter list parameters are not supported for SPACRLSE:

- GDGVER
- VALUE

Usage Notes

SPACRLSE provides services for DASD physical sequential data sets. Unused space is released from the data set when the data set is closed. SPACRLSE does not release the space itself, but sets the RLSE JCL parameter for selected resources.

SPACRLSE works for both new allocations and existing data sets.

Note: SPACRLSE does not work on data sets allocated through ISPF service 3.2 (data set utilities).

The RLSE parameter specifies the scope of the function. *ALL* specifies that all selected data sets should be flagged for space release. *SEC* specifies that only data sets with a secondary allocation should be flagged for release; data sets with only a primary allocation are ignored. This allows the primary allocation to remain at full size, which may be desirable if the data set is extended at a later time.

The RLSE parameter must be specified; SPACRLSE takes no action without a rule specifying the release scope for selected resources.

SPACRLSE does not modify the action of OS/390 in processing the release of unused data set space; therefore, OS/390 restrictions still apply. See the discussion of the RLSE subparameter of the SPACE parameter in the *JCL Reference Manual* for your release of OS/390.

Note that the SPACRLSE function must be specified in the SMFUNCxx member to be available and must be set active to provide services.

Messages

If RLSE=SEC or ALL, the message issued by SPACRLSE is:

SVM3349I *job,step,dd,dsn* RELEASE PARAMETER ASSIGNED

If RLSE=NO, the message issued by SPACRLSE is:

SVM3349I *job,step,dd,dsn* RELEASE PARAMETER REMOVED

Example

This example shows you how to release space for all data sets except certain temporary work files which are frequently extended during the course of a job; for production jobs, do not release space in data sets with only a primary allocation.

SMFUND5 member

```
SET  NAME=SPACRLSE
      ACTIVE=YES
      MSG=I SMF=I
      FLST=D3 RLST=D3
      DESC='FREE UNUSED SPACE'
```

The function is defined and activated.

SMVARSD0 member

```
SET  VARIABLE=DLYWORK
      VALUE=**.DLYWORK*
      VALUE=**TRAN*./
      VALUE=**TEMP*
      VALUE=WORK/
```

Define a variable which identifies all transitory work files that should not have free space deleted.

SMFLSTD3 member

```
SET  MODE=ACT
      EXC DSN=&DLYWORK
      INC DSN=
```

Select all data sets for SPACRLSE processing, except those specified in the variable &DLYWORK.

SMRLSTD3 member

SET RLSE=SEC
INC JOBCLASS=P

For production jobs, release space only if the data set has a secondary allocation.

SET RLSE=ALL
INC DSN=

For all other data sets, release space.

SPACSQTY—Set Primary and Secondary Space for Data Set

The SPACSQTY function sets primary and secondary space allocation values for non-VSAM and VSAM DASD data sets being created. Exceptions for VSAM allocations are noted in the parameters explanations.

Primary and secondary space allocation values have a significant role in effective use of DASD space. Allocations which are consistently too high, even for temporary data sets, utilize more space than is necessary, possibly preventing other jobs from allocating needed resources. Allocations which are too low require subsequent processing of some type, either automated or manual, to allocate more space to the data set.

SPACSQTY allows the data center to set requirements and limitations on data set size and to enforce those requirements rigorously. Application of predefined primary and secondary space allocations can be based on data set names, on device types, on assigned pools, on size or maximum possible size, or on any property allowed as a MAINVIEW SRM filter list parameter.

Rule List Parameters

SET Statement (in member SMRLSTxx)

Parameter	Description
ALCTYPE=xxx	Quantity unit of space allocation (TRK, CYL, KAV, MAV, UAV, BLK, KB, MB) that applies to both primary and secondary space quantities
AVL=nnnnn	Average block or record length
CONTIG=YES/NO	Specifies whether a data set is allocated with contiguous space required
DIR=nnnn	Sets number of directory blocks for partitioned data sets
EVENTID=xxxxx	Identifies a user event defined in an SMEVNTxx member
PQTY=nnnnn	Primary space allocation in unit specified
REPLACE=YES/NO	Allows override to JCL-specified space allocations
ROUND=YES/NO	Rounds space allocated in blocks to nearest cylinder
SQTY=nnnnn	Secondary space allocation in unit specified

Rule List Parameter Explanations**ALCTYPE=**

Purpose Specifies the unit of space allocation. Applies to both primary and secondary space quantities.

Syntax: ALCTYPE=*xxx*

where *xxx* is a valid allocation type from the following list:

KB	a block allocation where PQTY and SQTY are assumed to be kilobytes
MB	a block allocation where PQTY and SQTY are assumed to be megabytes
TRK	tracks
CYL	cylinders
KAV	AVGREC=K
MAV	AVGREC=M
UAV	AVGREC=U
BLK	blocks

Note: KAV, MAV, and UAV are not valid parameters for VSAM allocations.

Default: KB

AVL=

Purpose Specifies the average block size for BLK allocations and record length for KAV, MAV, and UAV allocations.

Syntax: AVL=*nnnnn*

where *nnnnn* is a number in the range 1 – 32,767.

Default: 9000

CONTIG=

Purpose: Specifies whether a data set is allocated with contiguous space required.

Syntax: CONTIG=*YES/NO*

Default: None

DIR=

Purpose: Specifies the number of directory blocks for partitioned data sets.

Note: This parameter can only be used to change the directory blocks. The user must request DSORG=PO or directory blocks in the original allocation.

Syntax: DIR=*nnnn*

where *nnnn* is a number between 1 and 4096.

Default: None

EVENTID=

Purpose: Specifies the identifier assigned to a user event in SMEVNT_{xx}. This parameter will cause an event to be generated from this function.

Note: If EVENTID= is used on an FLST SET statement with MODE=INACT, the event will still be issued.

Syntax: EVENTID=*xxxxx*

where *xxxxx* is the 5-character string specified on the EVNTID parameter in SMEVNT_{xx}.

Default: None

PQTY=

Purpose: Specifies the primary space allocation quantity in the unit specification ALCTYPE for a new non-VSAM data set.

Syntax: PQTY=*nnnnn*

where *nnnnn* is a number in the range 1–99999.

Default: None

REPLACE=

Purpose: Specifies whether the space allocations already specified for the data set (in a JCL, for example) are overridden. If REPLACE=NO, and a primary or a secondary space allocation already exists for the new data set, then the SPACSQTY function does not replace either space allocation value. If REPLACE=YES, the SPACSQTY function sets the primary and secondary space allocations, regardless of any existing space parameters.

Syntax: REPLACE=YES/NO

Default: REPLACE=NO

Note: REPLACE only applies to PQTY and SQTY parameters. CONTIG and ROUND are not affected by REPLACE.

ROUND=

Purpose: Specifies that the space allocation be rounded to the nearest cylinder for new data sets.

Note that this rounding process is performed by OS/390; MAINVIEW SRM merely switches on the appropriate bit to allow OS/390 to perform this service.

Syntax: ROUND=YES/NO

Default: None

Note: Both SPACCONV and SPACSQTY functions allow the ROUND parameter to be specified. If both are specified for the same data set, the value in SPACCONV will be the value assigned to the data set, because SPACCONV receives control after SPACSQTY.

SQTY=

Purpose: Specifies the secondary space allocation quantity in the units specified in ALCTYPE for a new non-VSAM data set.

Syntax: SQTY=nnnnn

where *nnnnn* is a number in the range 1–99999.

Default: None

Parameters Not Supported

The following filter list parameters are not supported for SPACSQTY:

- GDGVER
- VALUE

Usage Notes

SPACSQTY sets the data set's primary and secondary space allocation in the units specified in the ALCTYPE parameter.

Note: For compatibility with earlier MAINVIEW SRM releases, PQTY and SQTY are assumed to be in kilobytes if ALCTYPE is not coded. The resulting allocation will be calculated in average blocks based on the AVL parameter.

SPACSQTY provides services for VSAM and non-VSAM DASD data sets being created; old data sets are not affected by SPACSQTY

Note: ALCTYPE applies to both primary and secondary space quantities. If, for example, PQTY and ALCTYPE are set for an allocation, ALCTYPE also applies to the secondary that was originally assigned to the allocation.

The control parameters (ALCTYPE, AVL, CONTIG, DIR, ROUND) allow full control over space allocation specifications. ALCTYPE specifies the unit of space that is referred to by PQTY and SQTY; for example, PQTY=10 ALCTYPE=CYL requests 10 cylinders of space.

The SPACSECA function also allows specification of secondary allocation quantity. SPACSECA, however, is invoked when a data set is out-of-space on its primary extent.

Note: The ISPF data set information display in functions 3.4 and 3.2 do not always give the correct values for secondary space in blocks, particularly when the data set is allocated without a blocksize. The information provided by IEHLIST is correct, however. Note that the SPACSQTY function must be specified in the SMFUNCxx member to be available and must be set active to provide services.

REPLACE parameter

The REPLACE parameter allows SPACSQTY to be applied in either a restrictive or a lenient manner. The REPLACE parameter specifies whether a previously-specified space allocation is to be overridden by the SPACSQTY PQTY and SQTY parameters.

- REPLACE=YES forces the use of the space allocation established in SPACSQTY, regardless of any space parameters already specified in the JCL or elsewhere.
- REPLACE=NO accepts the original space parameters. The space allocation set by SPACSQTY is applied only if there is no existing space allocation specification.

Messages

SPACSQTY generates message SVM3446I when it provides new space allocation values for a data set:

```
SVM3446I  job,step,dd,dsn SPACE SET TO: TYPE=alloc_type,  
          PRIMARY=prim_alloc, SECONDARY=sec_alloc, DIRECTORY  
          BLOCKS=dir_blocks, AVL=avl_length
```

Messages issued by SPACSQTY include:

```
SVM3317I  job,step,dd,dsn CONTIG y/n ASSIGNED  
SVM3318I  job,step,dd,dsn ROUND y/n ASSIGNED
```

Controlling Extent Size

In general, extent-size during allocation/volume-addition can be controlled by several StopX37/II functions when at least one of the eligible volumes contain an extent large enough to satisfy the requested size or when SPACSWIR is active, and none of the eligible volumes contain an extent large enough to satisfy the requested size. The StopX37/II functions are:

- SPACSQTY adds or overrides requested space on new allocations.
- SPACPRIM reduces the size of the primary extent if the requested value cannot be found.
- SPACSWIR reduces the size of the first allocation on an added volume if the requested value cannot be found.
- SPACVOLA adds a volume when OS/390 encounters an end-of-volume condition:

Examples

The following example shows you how to force a primary and secondary space allocation limit on all new data sets being allocated on volumes starting with P and whose data set name ends with TEST, and with a normal termination disposition of DELETE.

SMFUNCxx member

```
SET  NAME=SPACSQTY ACTIVE=YES
      MSG=I SMF=E
      FLST=JV RLST=JV
      DESC='SET PRIM AND SEC SPACE QTY'
```

The function is defined and activated.

SMFLSTJV member

```
SET  MODE=ACT
      INC LLQ=TEST/ VOL=P/
```

This filter list specification is active. It selects all data sets whose final qualifier begins with TEST, and which are being allocated to any volume whose name begins with P.

SMRLSTJV member

```
SET  PQTY=100 SQTY=20KB REPLACE=YES  X
      ALCTYPE=KB
      INC DISP2=DELETE
```

All data sets selected by the FLST specification above and with a normal termination disposition of DELETE are given a primary and secondary space allocation as shown. Any space allocation parameters in the JCL are overridden.

The following example shows you how to:

- impose space limitations determined by the fourth account parameter of the job step EXEC card for all data sets going on a public work device (POOL=PUBWRK), except for those jobs with a third job account parameter beginning with PR
- apply this limitation in simulation mode only, to determine impact

SMFUNCxx member

```
SET  NAME=SPACSQTY ACTIVE=YES
      MSG=I SMF=I
      FLST=23 RLST=22
      DESC='SET PRIM AND SEC SPACE QTY'
```

The function is defined and activated.

SMFLST23 member

MODE=SIM
INC POOL=PUBWRK
EXC JOBACCT3=PR/

This filter list specification is in simulation mode only. It selects all data sets in pool PUBWRK.

SMRLST22 member

SET PQTY=1000 SQTY=500 REPLACE=YES ALCTYPE=KB
INC STEPACCT3=LRG

Force a primary and secondary space allocation on data sets selected by the filter list above, depending on the value in the third account field of the step EXEC statement.

SET PQTY=500 SQTY=100 REPLACE=YES ALCTYPE=KB
INC STEPACCT3=MOD

SET PQTY=100 SQTY=25 REPLACE=YES ALCTYPE=KB
INC STEPACCT3=SML

SET PQTY=50 ALCTYPE=CYL CONTIG=YES
REPLACE=YES
INC DSN=ACCT/ DSORG=PS SIZE<5MB

Allocate 50 cylinders for all PS accounting data sets that are allocated with less than 50 MB.

STKSUPP—Split Unit Affinity Specifications Over Devices

The STKSUPP function splits unit affinity specifications across separate devices to eliminate requirements to enter or remove volumes from silos. STKSUPP also allows specification of default units for scratch volumes outside of silos.

When unit affinity JCL parameters force multiple DDs onto a single drive, volumes may be required to be entered into or removed from silos. HSCs software component determines the location of the volume specified in the *primary* DD statement; volumes on *secondary* DD statements have to be moved to use the unit allocated to the affinity group.

STKSUPP alleviates the problem by splitting the affinity into groups based on tape location—for example, into a cartridge drive group and a silo group. While the affinity split requires more drives to run the job, it does not require operator intervention to enter or remove volumes from silos.

System Parameters

The system parameter that affects STKSUPP is described in Table 3-12.

Table 3-12 **System Parameters—STKSUPP**

Parameter	Purpose
STKSCR	Specifies the default location of scratch tapes for the STKSUPP function.

Rule List Parameters

SET Statement (in member SMRLST_{xx})

Parameter	Description
DEFUNIT=xxxxxxx	Generic unit name for volumes outside the silos
EVENTID=xxxxx	Identifies a user event defined in an SMEVNT _{xx} member
SPLIT= <u>YES</u> /NO	Specifies whether to split unit affinities with STK silos

Rule List Parameter Explanations**DEFUNIT=**

Purpose: Specifies the default unit name for volumes located outside STK silos. Under some circumstances with silos in separate physical locations, the HSC processing can cause a non-silo tape to be allocated to a silo device in a remote location, possibly without an operator. The DEFUNIT parameter forces non-silo volumes to be allocated to a local silo.

Syntax: DEFUNIT=xxxxxxx
where xxxxxx specifies a generic unit name.

Default: DEFUNIT=3480

EVENTID=

Purpose: Specifies the identifier assigned to a user event in SMEVNTxx. This parameter will cause an event to be generated from this function.

Note: If EVENTID= is used on an FLST SET statement with MODE=INACT, the event will still be issued.

Syntax: EVENTID=xxxxx
where xxxxx is the 5-character string specified on the EVNTID parameter in SMEVNTxx.

Default: None

SPLIT=

Purpose: Specifies whether affinity separation should be applied to resources selected for the STKSUPP function. When unit affinity directs multiple DD statements to a single drive, STKSUPP can apply affinity separation to allocate separate devices for silo and non-silo volumes, so volumes would not have to entered into or removed from silos.

Syntax: SPLIT=YES/NO

Default: SPLIT=YES

Parameters Not Supported

The following filter list parameters are not supported for STKSUPP:

ALCTYPE	BUFSP	CATALOG
CONTIG	CURSPACE	DSORG
ENVIR	ERASE	EXTENT
GDGVER	IMBED	MAXSIZE
OWNER	PRISPACE	RECORG
REPL	REUSE	ROUND
SECSPACE	SIZE	VALUE
VSAMCOMP	VSAMDEF	

Usage Notes

When unit affinity is used to group DD statements onto a single drive in conjunction with StorageTek tape silos, the first *primary* DD statement's drive is allocated for the group, and volumes on subsequent DD statements are required to use that drive. This can require tapes to be entered into or removed from tape silos. For example:

```
//STEP10 EXEC  PGM=....  
//DD1      DD   DSN=DSN1,UNIT=CART,VOL=SER=SILOA1,DISP=OLD  
//DD2      DD   DSN=DSN2,UNIT=AFF=DD1,VOL=SER=TAPE01,DISP=OLD  
//DD3      DD   DSN=DSN3,UNIT=AFF=DD1,VOL=SER=SILOB1,DISP=OLD
```

HSC requires OS/390 to select silo A (for tape SILOA1); the affinities on DD2 and DD3 then require the non-silo tape TAPE01 and the silo B tape SILOB1 to be entered into silo A.

The STKSUPP function can be directed to split the affinities so that tapes are not required to be relocated in or out of silos. For example:

```
SET  SPLIT=YES  
INC  UNIT=CART
```

With this specification in the rule list for STKSUPP, the above JCL would require 3 devices: silo A for DD1, a cartridge drive for DD2, and silo B for DD3. This is the equivalent of:

```
//STEP10 EXEC PGM=....
//DD1 DD DSN=DSN1,UNIT=SILOA,VOL=SER=SILOA1,DISP=OLD
//DD2 DD DSN=DSN2,UNIT=CART,VOL=SER=TAPE01,DISP=OLD
//DD3 DD DSN=DSN3,UNIT=SILOB,VOL=SER=SILOB1,DISP=OLD
```

STKSUPP does not remove unit affinity from devices within the same silo or from *manual* drive groups. For example:

```
//STEP10 EXEC PGM=....
//DD1 DD DSN=DSN1,UNIT=CART,VOL=SER=SILOA1,DISP=OLD
//DD2 DD DSN=DSN2,UNIT=AFF=DD1,VOL=SER=TAPE01,DISP=OLD
//DD3 DD DSN=DSN3,UNIT=AFF=DD2,VOL=SER=SILOB1,DISP=OLD
//DD4 DD DSN=DSN4,UNIT=AFF=DD3,VOL=SER=TAPE02,DISP=OLD
//DD5 DD DSN=DSN5,UNIT=AFF=DD4,VOL=SER=SILOB2,DISP=OLD
//DD6 DD DSN=DSN6,UNIT=AFF=DD5,VOL=SER=SILOA2,DISP=OLD
```

Without affinity separation processing by STKSUPP, tapes TAPE01 and TAPE02 would have to be entered into silo A, and tape SILOB1 and SILOB2 would have to be ejected from silo B and entered into silo A (tapes SILOA1 and SILOA2 are already in silo A). STKSUPP separates affinity groups for silo A, silo B, and a cartridge drive, as if the following JCL had been coded:

```
//STEP10 EXEC PGM=....
//DD1 DD DSN=DSN1,UNIT=SILOA,VOL=SER=SILOA1,DISP=OLD
//DD2 DD DSN=DSN2,UNIT=CART,VOL=SER=TAPE01,DISP=OLD
//DD3 DD DSN=DSN3,UNIT=SILOB,VOL=SER=SILOB1,DISP=OLD
//DD4 DD DSN=DSN4,UNIT=AFF=DD2,VOL=SER=TAPE02,DISP=OLD
//DD5 DD DSN=DSN5,UNIT=AFF=DD3,VOL=SER=SILOB2,DISP=OLD
//DD6 DD DSN=DSN6,UNIT=AFF=DD1,VOL=SER=SILOA2,DISP=OLD
```

Scratch Tapes

STKSUPP cannot determine the appropriate location for scratch tapes. The STKSCR system parameters specified in the SMMSYSxx parmlib member define default locations for the following scratch requests: standard-label tapes, non-labeled tapes, ASCII tapes, and non-standard-label tapes.

Each type of scratch request can be directed to any silo (operand IN), a specific silo ID (*nnn*), or to devices outside of silos (operand OUT). For example:

```
STKSCR=( IN, OUT, OUT, OUT )
```

directs standard-label scratch tapes to silo devices and all other scratch tapes to non-silo devices.

In the following job, without STKSUPP and the preceding STKSCR parameters, all tapes would have to be moved to the device associated with DD1.

```
//STEP10 EXEC PGM=...
//DD1   DD   DSN=DSN1,UNIT=CART,DISP=NEW,LABEL=(1,SL)
//DD2   DD   DSN=DSN2,UNIT=AFF=DD1,DISP=NEW,LABEL=(1,NL)
//DD3   DD   DSN=DSN3,UNIT=AFF=DD2,DISP=NEW,LABEL=(1,NL)
//DD4   DD   DSN=DSN4,UNIT=AFF=DD3,DISP=NEW,LABEL=(1,SL)
//DD5   DD   DSN=DSN5,UNIT=AFF=DD4,VOL=SER=SILOB1,DISP=OLD
```

With the STKSCR parameters above, affinity separation splits the scratch requests into two groups: all standard-label scratch tapes and all other scratch tapes. It also splits out DD5, since it resides in silo B. This is equivalent to the following JCL:

```
//STEP10 EXEC PGM=...
//DD1   DD   DSN=DSN1,UNIT=SILOX,DISP=NEW,LABEL=(1,SL)
//DD2   DD   DSN=DSN2,UNIT=CART,DISP=NEW,LABEL=(1,NL)
//DD3   DD   DSN=DSN3,UNIT=AFF=DD2,DISP=NEW,LABEL=(1,NL)
//DD4   DD   DSN=DSN4,UNIT=AFF=DD1,DISP=NEW,LABEL=(1,SL)
//DD5   DD   DSN=DSN5,UNIT=AFF=SILOB,VOL=SER=SILOB1,DISP=OLD
```

Default Unit Names

Non-silo tapes allocated to silo devices can be allocated to remote silos, if your data center has silos that are physically separate. When OS/390 attempts to allocate a cartridge device and the HSC software changes the allocation to a silo, the original unit satisfies the request. OS/390 may then select a device in a remote silo, which may not have an operator.

The DEFUNIT rule list parameter on the STKSUPP rule list can override the unit specified for a non-silo tape request, ensuring that cartridges allocated to a silo are allocated to a local device. For example:

```
SET SPLIT=YES DEFUNIT=SILOB
INC UNIT=CART
INC UNIT=3480
```

With this specification, any non-silo tape request with CART or 3480 specified in the JCL will be allocated to silo B (any device represented by the generic name SILOB).

Messages

STKSUPP issues message SVM3600I when affinity processing for the specified ddname is changed.

SVM3600I AFFINITY MODIFIED ON *ddname* + *nnnnn* FROM *nnnn(nn)* TO *nnn*

STKSUPP issues message SVM3601I when unable to split the affinity due to a shortage of devices.

SVM3601W AFFINITY SPLIT ON *ddname* WOULD EXCEED AVAILABLE DEVICES

Example

This example shows you how to split affinities to prevent tape movement in and out of silos.

SMMSYS00 member

SET STKSCR=(IN,OUT,OUT,OUT)

Standard-label scratch tapes are allocated inside silos; all other scratch tapes are allocated outside silos.

SMFUNC00 member

SET FUNC=STKSUPP ACTIVE=YES
MSG=I SMF=N
FLST=25 RLST=25
DESC='STK UNIT AFFINITY SPLIT'

The function is active.

SMFLST25 member

SET MODE=ACT
INC UNIT=TAPE
INC UNIT=CART
INC UNIT=SILO*

All allocations coded for TAPE, CART, or any unit starting with SILO are selected for processing.

SMRLST25 member

SET SPLIT=NO
INC JOB=PAYR* JOBCLASS=V

Do not split affinities for payroll jobs in class V.

SET SPLIT=YESDEFUNIT=CART
INC UNIT=TAPE
INC UNIT=CART
INC UNIT=SILO*

Otherwise, separate affinities for allocations requesting TAPE, CART, or SILO units.

SUPJSCAT—Suppress JOBCAT and STEPCAT DD Statements

The SUPJSCAT function suppresses the action of all JOBCAT and STEPCAT DD statements in a job.

JOBCAT and STEPCAT statements allow circumvention of the cataloging structure implemented by the data center. Jobcat/stepcats allow data sets to be cataloged in a catalog that does not have an entry in the OS/390 master catalog. These data sets are then accessible only through a jobcat/stepcat in the JCL (or a hard-coded volume ID); they cannot be located by the normal OS/390 catalog search process. This can create problems if access is needed to the data sets by personnel who are not aware of the jobcat/stepcat requirements.

JOBCATs and STEPCATs are not supported by DFSMS; use of these DD statements in an DFSMS-managed system causes a JCL error.

SUPJSCAT provides a method of monitoring jobs for jobcat/stepcat usage without causing problems (in simulation mode) and makes JCL changes unnecessary for the removal of jobcat/stepcat cards.

Rule List Parameters

SET Statement (in member SMRLSTxx)

Parameter	Description
EVENTID=xxxxx	Identifies a user event defined in an SMEVNTxx member

Rule List Parameter Explanations

EVENTID=

Purpose: Specifies the identifier assigned to a user event in SMEVNTxx. This parameter will cause an event to be generated from this function.

Note: If EVENTID= is used on an FLST SET statement with MODE=INACT, the event will still be issued.

Syntax: EVENTID=xxxxx

where xxxxx is the 5-character string specified on the EVNTID parameter in SMEVNTxx.

Default: None

Usage Notes

SUPJSCAT provides jobcat/stepcat suppression for all selected jobs.

Except for EVENTID, there are no rule list parameters for SUPJSCAT. Resource selection is accomplished with the filter list.

Due to the point at which SUPJSCAT is invoked, only the following filter list parameters are available:

DD	DSN	DSNn
HLQ	JOB	LLQ
NQUAL	STEP	UNIT
FUNCTION	RACF	USER
USRCn	USRNn	

Messages

The message issued by SUPJSCAT when a JOBCAT or STEPCAT is suppressed is:

SVM0420I *job,step,dd,dsn* STEPCAT/JOBCAT SUPPRESSED

Note that the SUPJSCAT function must be specified in the SMFUNCxx member to be available and must be set active to provide services.

Example

This example shows you how to suppress jobcat/stepcat cards immediately on test jobs and in simulate mode for production jobs. Started task jobs are exempt, as are jobs submitted by system programmers.

SMFUNC51 member

```
SET  NAME=SUPJSCAT ACTIVE=YES
      MSG=I SMF=I
      FLST=49
      DESC='OPEN EMPTY DATA SETS'
```

The function is defined and activated. Note that there is no RLST parameter.

SMFLST49 member

```
SET  MODE=INACT
      INC XMODE=STC
      INC USER=SYSP*
```

Exclude started tasks and system-programmers' jobs from SUPJSCAT processing (by including these resources for inactive mode—this could also have been accomplished by excluding these resources from all SET statements below).

SET	MODE=SIM	Simulate job/stepcat suppression
	INC JOBCLASS=P	for production jobs.
SET	MODE=ACT	Suppress job/stepcats for all other
	INC JOB=	jobs.

SUPVOLRF—Suppress Volume References

The SUPVOLRF function suppresses all DASD volume references.

Volume references (VOL=REF= on the DD statement in the JCL) prevent extension of a data set onto additional volumes by the MAINVIEW SRM function SPACVOLA. This restriction can result in unnecessary abends for out-of-space situations.

For each job step, SUPVOLRF examines all DASD data set allocations and removes the volume references for data sets selected by the filter list. It also removes any UNIT=AFF specifications.

Rule List Parameters

SET Statement (in member SMRLSTxx)

Parameter	Description
EVENTID=xxxxx	Identifies a user event defined in an SMEVNTxx member

Rule List Parameter Explanations

EVENTID=

Purpose: Specifies the identifier assigned to a user event in SMEVNTxx. This parameter will cause an event to be generated from this function.

Note: If EVENTID= is used on an FLST SET statement with MODE=INACT, the event will still be issued.

Syntax: EVENTID=xxxxx

where xxxxx is the 5-character string specified on the EVNTID parameter in SMEVNTxx.

Default: None

Usage Notes

SUPVOLRF operates on any DASD data sets, new or existing. SUPVOLRF will not operate on TAPE data sets.

Prior to DFSMS 1.3, allocations that included the VOL=REF parameter were not processed by ACS routines. For new allocations, if the referenced data set was SMS-managed, the storage class of the referenced data set was assigned to a new data set and the new data set became SMS-managed. If the referenced data set was not SMS-managed, the new data set also became non-managed. The SUPVOLRF function can be used to ensure all new allocations are processed through ACS, and the possibility of creating a non-managed data set by simply referring to a non-managed data set is avoided.

Effective with DFSMS 1.3, allocations with VOL=REF coded *are* processed by ACS routines. The volume serial of the referenced data set is not provided in the ACS read-only variables &ANYVOL or &ALLVOLS; rather, pseudo volume serial numbers are used to indicate the type of data set referenced. The VOL filter list parameter will contain one of the following values in the SMSACSDC, SMSACSMC, SMSACSSC, SMSACSSG, and SMSACSTE functions if the SUPVOLRF function *does not* suppress the VOL-REF:

REF=NS	the referenced data set is not SMS-managed
REF=ST	the referenced data set is an SMS-managed tape file
REF=SD	the referenced data set an SMS-managed DASD or VIO file

If SUPVOLRF does suppress the VOL=REF, none of the above pseudo volume serial numbers will be found in the ACS read-only variables &ANYVOL or &ALLVOLS nor in the VOL filter list parameter.

If VOL=REF=dsname refers to an uncataloged data set, the job will normally fail with a JCL error. If the VOL=REF is suppressed by the SUPVOLRF function, the JCL error due to the reference to an uncataloged data set will be avoided. However, if the allocation is for an existing data set, a JCL error may still occur; for example, if the data set is not cataloged, no volume serial information is available to complete the allocation. If the allocation is for a new data set, removing the VOL=REF will make the new allocation non-specific.

The volume serial of the referenced data set is available in the REFVOL filter list parameter during SMSACSDC, SMSACSMC, SMSACSSC, SMSACSSG, and SMSACSTE functions. The REFVOL is also available during the DASDPOOL function. In the case of a reference to an uncataloged data set as discussed above, the REFVOL will contain the value NULVRF.

Because the function has no RLST operands to set, all resource selection must be done through the FLST.

Due to the point at which SUPVOLRF is invoked, only the following filter list parameters are available:

DD	DSN	DSNn
FUNCTION	HLQ	JOB
LLQ	NQUAL	STEP
UNIT	USRCn	USRNn

Note that the SUPVOLRF function must be specified in the SMFUNCxx member to be available and must be set active to provide services.

SUPVOLRF generates message SVM0421I when it removes a volume reference:

SVM0421I *job,step,dd,dsn* VOL=REF SUPPRESSED

Examples

The following example shows you how to suppress volume references for all Accounts Payable (production) data sets. (In this example, Accounts Payable data sets are identified by a high-level qualifier of DIVA and a third qualifier that starts with TR.)

SMFUNCxx member

```
SET  NAME=SUPVOLRF  ACTIVE=YES  The function is defined and activated.
      MSG=I SMF=N      There is no RLST specification. Do not
      FLST=34          produce any SMF records.
      DESC='SUPPRESS VOL=REF'
```

SMFLST34 member

```
SET  MODE=ACT      This filter list specification is active.
      INC HLQ=DIVA DSN3=TR*  Selected resources are all data sets
                              with a high-level qualifier of DIVA and
                              a 3rd qualifier beginning with TR.
```

The following example shows you how to suppress volume references for a group of frequently selected data set resources identified by a MAINVIEW SRM variable.

SMMSYS10 member

```
SET  FUNC=11  VARS=55  POOL=11  The MAINVIEW SRM system is
      CMDPREF=#  SMFID=251      defined.
      PASSWORD=123456789012
```

SMFUNC11 member

SET NAME=SUPVOLRF ACTIVE=YES MSG=I SMF=I FLST=02 DESC='SUPPRESS VOL=REF'	The function is defined and activated. There is no RLST specification.
---	---

SMFLST02 member

SET MODE=ACT INC DSN=&DSNEXT	This filter list specification is active. It selects data sets specified in the variable DSNEXT.
-----------------------------------	--

SMVARS55 member

SET VARIABLE=DSNEXT VALUE=AP*.*.MSTR VALUE=GL035B.TRAN/ VALUE=AR*.*.DLYUPD VALUE=FLS*/ VALUE=TPROV?.*.WKLY./	The variable DSNEXT is defined to include the list of data set name masks.
---	--

TAPECOMP—Control Tape Data Set Compaction (IDRC)

The TAPECOMP function specifies the use of data compaction for a data set on a tape cartridge. Improved Data Recording Capability (IDRC) must be available on the magnetic tape subsystem.

Data compaction on appropriately equipped tape devices can increase the effective capacity by up to 5 times and boost the effective data rate by as much as 70%. This provides the capability of greatly increased overall storage capacity for tape data sets, if it is used consistently. OS/390 requires specification of TRTCH=COMP in the DCB parameters on the DD statement in JCL or the DEVSUPxx member of SYS1.PARMLIB to activate data compaction.

TAPECOMP allows the data center to activate data compaction without changes to JCL. Specification of tape compaction usage can be based on data set names, on device types, on assigned pools, or on any property allowed as a MAINVIEW SRM filter list parameter.

Rule List Parameters

SET Statement (in member SMRLSTxx)

Parameter	Description
COMP=YES/NO	Specifies the use of data compaction
EVENTID=xxxxx	Identifies a user event defined in an SMEVNTxx member

Rule List Parameter Explanations

COMP=

Purpose: Specifies whether data compaction should be used for a data set allocation on a tape cartridge.

Syntax: COMP=YES/NO

Default: None

EVENTID=

Purpose: Specifies the identifier assigned to a user event in SMEVNTxx. This parameter will cause an event to be generated from this function.

Note: If EVENTID= is used on an FLST SET statement with MODE=INACT, the event will still be issued.

Syntax: EVENTID=xxxxx

where xxxxx is the 5-character string specified on the EVNTID parameter in SMEVNTxx.

Default: None

Parameters Not Supported

The following filter list parameters are not supported for TAPECOMP:

ALCTYPE	BUFSP	CATALOG
CONTIG	CURSPACE	DSORG
ENVIR	ERASE	EXTENT
GDGVER	IMBED	MAXSIZE
OWNER	PRISPACE	RECORG
REPL	REUSE	ROUND
SECSPACE	SIZE	VALUE
VSAMCOMP	VSAMDEF	

Usage Notes

TAPECOMP provides services only for tape data sets.

TAPECOMP is not able to verify that the tape device truly supports IDRC.

Note that the TAPECOMP function must be specified in the SMFUNCxx member to be available and must be set active to provide services.

Messages

TAPECOMP generates message SVM3452I when it processes a data set:

SVM3452I *job,step,dd,dsn* TAPE COMPACTION SET TO *on/off*

Example

This example shows you how to activate data compaction for all tape data sets on 3490 devices with IDRC (defined in a pool named 3490C).

SMMSYS10 member

```
SET    FUNC=21 POOL=31  
        CMDPREF=# SMFID=251  
        PASSWORD=123456789012
```

The MAINVIEW SRM system is defined.

SMFUNC21 member

```
SET    NAME=TAPECOMP  
        ACTIVE=YES  
        MSG=E SMF=N  
        FLST=02    RLST=02  
        DESC='COMPACTION FOR  
        TAPES'
```

The function is defined and activated. SMF records are not written. Only error messages are written.

SMPOOL31 member

```
SET    POOLNAME=3490C  
        TYPE=3490  
        INC ADR=180  
        ADR=181  
        ADR=182  
        ADR=183
```

The pool 3490C is defined to include a list of 3490 devices.

SMFLST02 member

```
SET    MODE=ACT  
        INC POOL=3490C
```

This filter list specification is active. It selects all data sets going to pool 3490C.

SMRLST02 member

```
SET    COMP=YES  
        EXC DISP2=DELETE
```

This rule list specifies data compaction for all data sets whose normal termination disposition is not DELETE.

TAPedefr—Defer Tape Mounts

The TAPedefr function causes OS/390 to defer the tape mount request until the data set is opened.

Data sets allocated to tape devices normally cause a tape mount as soon as the data set is allocated (during job initialization). In many cases, this may be undesirable, since the data set may never be opened. Nevertheless, once the tape mount is issued, the job waits for the device to be readied.

TAPedefr allows the tape mount to be deferred until data set is opened for selected resources.

Rule List Parameters

SET Statement (in member SMRLSTxx)

Parameter	Description
EVENTID=xxxxx	Identifies a user event defined in an SMEVNTxx member

Rule List Parameter Explanations

EVENTID=

Purpose: Specifies the identifier assigned to a user event in SMEVNTxx. This parameter will cause an event to be generated from this function.

Note: If EVENTID= is used on an FLST SET statement with MODE=INACT, the event will still be issued.

Syntax: EVENTID=xxxxx

where xxxxx is the 5-character string specified on the EVENTID parameter in SMEVNTxx.

Default: None

Parameters Not Supported

The following filter list parameters are not supported for TAPedefr:

ALCTYPE	BUFSP	CATALOG
CONTIG	CURSPACE	DSORG
ENVIR	ERASE	EXTENT

GDGVER	IMBED	MAXSIZE
OWNER	PRISPACE	REORG
REPL	REUSE	ROUND
SECSpace	SIZE	VALUE
VSAMCOMP	VSAMDEF	

Usage Notes

TAPedefr defers the tape mount request by effectively adding the DEFER subparameter to the UNIT JCL parameter. Although the tape mount is deferred until the data set is opened, the tape device is still allocated at job initialization.

Since there are no rule list parameters for TAPedefr, there is no rule list. Resource selection is accomplished solely with the filter list.

The message issued by TAPedefr when a data set is marked for a deferred mount is:

```
SVM3454I  job,step,dd,dsn DEFER ASSIGNED
```

Note that the TAPedefr function must be specified in the SMFUNCxx member to be available and must be set active to provide services.

Example

This example shows you how to defer tape mounts on all jobs except certain high speed tape related production.

SMFUNC04 member

```
SET  NAME=TAPedefr
      ACTIVE=YES
      MSG=I SMF=I
      FLST=14
      DESC='DEFER TAPE MOUNTS'
```

The function is defined and activated.
Note that there is no RLST parameter.

SMFLST14 member

```
SET  MODE=ACT
      EXC PGM=TSET/
      EXC PGM=TDUP/
      INC JOB=/'
```

Exclude two sets of programs by partial name. For all other jobs, defer tape mounts until the data set is opened.

TAPEPOOL—Assign Tape Data Sets to Pools

The TAPEPOOL function assigns tape data sets to a pool of devices (including DASD devices), overriding any UNIT assignment in the JCL.

The ability to define pools of tape devices, and direct data sets to those pools, allows the storage administrator to more effectively manage the data center's tape resources by assigning usage based on flexible criteria. Long-duration jobs can be assigned to one pool, jobs requiring higher speed can be assigned to a pool containing only 3490 devices, jobs producing very large data sets can be assigned to a pool with IDRC support for compaction, etc.

TAPEPOOL allows data sets to be assigned to pools based on any of the selection criteria supported by MAINVIEW SRM, without reliance on JCL parameters. TAPEPOOL also allows data sets to be redirected from a tape device to a DASD device.

System Parameters

The system parameters that affect TAPEPOOL are described in Table 3-13.

Table 3-13 **System parameters—TAPEPOOL**

Parameter	Purpose
DMYUNIT	Defines the conversion of a nonexistent UNIT parameter to a valid UNIT parameter only if JCLEXT=YES.
REJECT	Controls termination of processing at the first or last data set rejected by REJECT=YES in DASDPOOL or DSNCHECK. If termination is to take place on the first rejected data set, code REJECT=FIRST. If all data sets are to be processed before control is returned to OS/390 allocation, code REJECT=LAST.
TAPEGENR	Specifies tape device generic names that some EasyPOOL functions will intercept. If you want to intercept all tape requests, specify ALLTAPE as the first generic name. (However, you cannot use ALLTAPE when JCLEXT=NO.) TAPEGENR affects functions DSNCHECK, FORCECAT, SETEXPDT, and TAPEPOOL.

Rule List Parameters

SET Statement (in member SMRLSTxx)

Parameter	Description
EVENTID=xxxxx	Identifies a user event defined in an SMEVNTxx member
NVOL=nn	Number of volumes and units that can be allocated to a data set.
POOL=(xxxx,xxxx, . . .)	Specifies one or more pools eligible for allocation
REJECT=YES/NO	Specifies whether the request should be rejected

Parameter	Description
UNIT=xxxxxxxx	Unit generic name (1–8 characters)
USEVOL=STOR/PRIV/ <u>ALL</u>	Directs volume allocation to a specific type (STOR, PRIV, ALL)

Rule List Parameter Explanations

EVENTID=

Purpose: Specifies the identifier assigned to a user event in SMEVNT $_{xx}$. This parameter will cause an event to be generated from this function.

Note: If EVENTID= is used on an FLST SET statement with MODE=INACT, the event will still be issued.

Syntax: EVENTID=xxxxx

where xxxxx is the 5-character string specified on the EVNTID parameter in SMEVNT $_{xx}$.

Default: None

NVOL=

Purpose: Specifies the number of volumes and units that can be allocated to a data set.

Syntax: NVOL= nn

where nn is a number in the range of 1–59 for non-VSAM data sets and 1–20 for VSAM data components.

Default: None

POOL=

Purpose: Specifies one or more pools (defined in SMPOOL $_{xx}$) to which the data set can be assigned. If the poolname is not defined, the refresh or start-up will fail. This is a required parameter if REJECT=YES is not specified.

Syntax: POOL=(xxxxxx,xxxxxx, . . .)

where xxxxxx specifies pool names from SMPOOL $_{xx}$. If a single pool is specified, the parentheses are not required.

Default: None

REJECT=

Purpose: Specifies that the request should be rejected; the allocation fails.

Syntax: REJECT=*YES/NO*

Default: REJECT=NO

UNIT=

Purpose: Specifies a unit name (esoteric, generic, or pool name) to be used by the data set during allocation.

Syntax: UNIT=*xxxxxxxx*

where *xx* is a valid unit name (esoteric, generic, or pool name)
1–8 characters long.

Default: None

USEVOL=

Purpose: Specifies the type of volume (storage, private, or either) that activates non-specific allocation requests.

Syntax: USEVOL=*STOR/PRIV/ALL*

STOR storage mounted volume
PRIV privately mounted volume
ALL volume of any mount type

Default: USEVOL=ALL

Parameters Not Supported

The following filter list parameters are not supported for TAPEPOOL:

BUFSP	CATALOG	ENVIR
ERASE	EXTENT	GDGVER
IMBED	OWNER	RECORG
REPL	REUSE	VALUE
VSAMCOMP	VSAMDEF	

Usage Notes

TAPEPOOL applies to data sets assigned to a tape device. The rule list can assign the data set to one or more eligible pools, or reject the allocation request entirely. The pools are defined in the SMPOOL_{xx} PARMLIB member.

TAPEPOOL does not change the device assignment if UNIT=AFF has been specified, or if the data set is a passed data set.

Unless REJECT=YES is specified, the POOL parameter must be specified; TAPEPOOL takes no action without a rule specifying a device pool to which selected resources may be assigned.

TAPEPOOL allows tape data set allocations to be directed to a pool of tape devices defined in SMPOOL_{xx} using the ADR parameter. Allocations to a tape generic or esoteric name, but not a specific unit address, will be allowed to one of the tape units in the pool.

TAPEPOOL allows a data set to be redirected from tape to DASD by specifying a DASD pool in the POOL rule list parameter. If this feature is used, SPACSQTY should also be used to provide space parameters for the DASD allocation; without some primary allocation amount, the DASD allocation for the redirected data set will fail. Note that if the JCL requests a specific tape volume (VOL=SER=xxx), the allocation is not interrupted and, therefore, is not redirected to a DASD pool.

TAPEPOOL ignores DFSMS-managed data sets.

TAPEPOOL does not support multiple pools. They can be specified, but the selected address can be from any one of the pools.

Messages

TAPEPOOL issues these messages:

When a pool is assigned:

SVM3338I *job,step,dd,dsn* ELIGIBLE POOL NAMES SET TO: pools

When an assignment is rejected:

SVM3311I *job,step,dd,dsn* REJECT = reject SET

Note that the TAPEPOOL function must be specified in the SMFUNC_{xx} member to be available and must be set active to provide services.

Example

This example shows you how to:

- define a tape pool for production jobs only
- direct data sets for jobs using UNIT=SYSTAPE to pools containing 3480 and 3490 devices
- reject tape data set allocation from users with a TD prefix on their IDs

The default tape pool should be 3420 devices.

SMPOOL05 member

SET	POOLNAME=T3420 TYPE=3420 INC ADR=(0340,0341)	Define a tape pool of 3420 devices.
SET	POOLNAME=T3480 TYPE=3480 INC ADR=(0380,0381,0382)	Define a tape pool of 3480 devices.
SET	POOLNAME=T3490A TYPE=3490 INC ADR=(0390,0391)	Define a tape pool of 3490 devices.
SET	POOLNAME=T3490B TYPE=3490 INC ADR=(039A,039B,039C,039D)	Define a second tape pool of 3490 devices.
SET	POOLNAME=REDIR TYPE=DASD INC VOL=(BNW002,BNW010)	Define a pool of DASD devices for redirection of tape data sets.

SMFUNC02 member

SET	NAME=TAPEPOOL ACTIVE=YES MSG=I SMF=N FLST=15 RLST=22 DESC='ASSIGN TAPE POOLS'	The function is defined and activated. The default for SMF records is none.
-----	--	---

SMFLST15 member

SET	MODE=ACT INC DSN=/	All data sets are selected for TAPEPOOL processing. (TAPEPOOL ignores allocations other than tape data sets.)
-----	-----------------------	---

SMRLST22 member

SET	REJECT=YES INC USER=TD/	Users with IDs beginning with TD are not allowed to run tape jobs.
SET	POOL=T3490B INC JOBCLASS=P	Production jobs are assigned to pool T3490B.
SET	POOL=(T3480,T3490A) INC UNIT=SYSTAPE	Any data sets originally assigned to the generic unit SYSTAPE will be assigned to a device in pool T3480 or T3490A.
SET	POOL=T3420 INC DSN=/	All other tape data sets are sent to the T3420 and T3480 pools.
SET	POOL=REDIR INC DSN=HR*.*.CLASS* INC DSN=HR*.*.PAY*	Certain files from the HR system are redirected to a DASD pool. Note that SPACSQTY must also be specified to provide space allocation information.

SMRLST25 member (for SPACSQTY)

SET PQTY=100 SQTY=10
INC POOL=REDIR

Use SPACSQTY to set a primary and a secondary allocation for all data sets redirected from tape to DASD.

VIOALLOC—Control VIO Allocation

The VIOALLOC function changes the allocation of temporary data sets to virtual storage (VIO).

The OS/390 VIO facility provides a significant advantage over normal DASD I/O. VIO resides in the paging space and I/O operations are performed at the speed of page access.

VIOALLOC changes the allocation of selected temporary data sets to (or from) VIO, with no JCL modifications.

Rule List Parameters

SET Statement (in member SMRLSTxx)

Parameter	Description
EVENTID=xxxxx	Identifies a user event defined in an SMEVNTxx member
VIO=YES/NO	Specifies whether the data set is allocated in VIO space

Rule List Parameter Explanations

EVENTID=

Purpose: Specifies the identifier assigned to a user event in SMEVNTxx. This parameter will cause an event to be generated from this function.

Note: If EVENTID= is used on an FLST SET statement with MODE=INACT, the event will still be issued.

Syntax: EVENTID=xxxxx

where xxxxx is the 5-character string specified on the EVNTID parameter in SMEVNTxx.

Default: None

VIO=

Purpose: Specifies whether the new data set will be allocated in VIO. This is a required parameter; VIO=YES or VIO=NO must be specified for any processing to occur.

Syntax: VIO=*YES/NO*

Default: None

Parameters Not Supported

The following filter list parameters are not supported for VIOALLOC:

- GDGVER
- VALUE

Usage Notes

VIOALLOC applies to new temporary data sets. VIOALLOC changes the allocation of selected data sets to VIO from DASD if the VIO parameter is YES, or to DASD from VIO if the VIO parameter is NO.

VIOALLOC only operates on temporary data sets (DSN=&&xxxx).

MAINVIEW SRM makes the data set eligible for VIO allocation, but OS/390 actually processes the request. Thus, OS/390 considerations apply to data sets switched to VIO. VIO allocation is discussed in “Data Set Resources—Allocation” in the *IBM OS/390 JCL User Guide*.

Messages

VIOALLOC issues two messages.

When a data set is directed to VIO:

SVM3439I *job,step,dd,dsn* MADE VIO ELIGIBLE

When a data set is prohibited from VIO:

SVM3440I *job,step,dd,dsn* VIO ELIGIBILITY REMOVED

If a data set is already allocated VIO and VIO=YES is specified, no action is taken and a message is not issued. Similarly, if a data set is not allocated VIO and VIO=NO is specified, no action is taken and a message is not issued.

Note that the VIOALLOC function must be specified in the SMFUNCxx member to be available and must be set active to provide service.

Example

This example shows you how to:

- specify VIO usage for all data sets with a maximum allocation of less than five megabytes
- ensure sort work data sets are not allocated to VIO

SMFUNC48 member

SET	NAME=VIOALLOC ACTIVE=YES	The function is defined and
	MSG=I SMF=I	activated.
	FLST=21 RLST=54	
	DESC='CONTROL VIO ALLOCATION'	

SMFLST21 member

SET	MODE=ACT	Select all data sets except those
	EXC DDNAME=SORT/	with sort-related ddnames.
	INC DSN=	

SMRLST54 member

SET	VIO=YES	Allow VIO for data sets whose
	INC MAXSIZE<5MB	maximum size is less than five
		megabytes.

VSAMCNTL—Enforce VSAM Allocation Standards

The VSAMCNTL function allows specification or suppression of various VSAM cluster definitions.

Some parameters that can be specified for VSAM file definitions can have significant affects on processing throughput. Parameters, such as ERASE and IMBED, may be undesirable except under certain circumstances. Other parameters, such as MASTERPW, can affect availability of the data and may be undesirable at your data center.

VSAMCNTL gives the storage administrator control over many of the sensitive VSAM definition parameters. VSAM definition parameters can be overridden for individual data sets or large groups of data sets, based on data set name, job class, user ID, and so on, giving the storage administrator complete control over critical VSAM specifications.

Rule List Parameters

SET Statement (in member SMRLSTxx)

Parameter	Description
BUFSP= <i>nnnnnn</i>	Buffer space for VSAM data sets (0–16776704)
CATALOG= YES/NO	Allows or removes IDCAMS CATALOG parameter
CISIZE=(<i>nnnnnn,nnnnnn</i>)	VSAM data and/or index control interval size (0–999999)
ERASE= YES/NO	Allows or removes IDCAMS ERASE parameter
EVENTID=xxxxx	Identifies a user event defined in an SMEVNTxx member
IMBED= YES/NO	Allows or removes IDCAMS IMBED parameter
OWNER=xxxxxxxxx	Assigns an owner ID to a VSAM cluster (1–40 characters)
PWDDEL= YES/ <u>NO</u>	Delete passwords specified in control statements
REPL= YES/NO	Allows or removes IDCAMS REPLICATE parameter
REUSE= YES/NO	Allows or removes IDCAMS REUSE parameter

Rule List Parameter Explanations**BUFSP=**

Purpose: Contains the buffer space for the cluster or the data component of VSAM data sets. Any existing buffer space specification is overridden.

Note: A BUFSP value that is too small will be ignored by VSAM. Generally, a value less than {2 x data CISIZE} for non-indexed files or a value less than {2x data CISIZE+1x index CISIZE} for indexed files will be too small.

Syntax: BUFSP=*nnnnnn*

where *nnnnnn* is a number in the range 0–16776704.

Default: None

CATALOG=

Purpose: Specifies the removal of the CATALOG parameter during cluster definition.

Syntax: CATALOG=*YES/NO*

If YES is specified the CATALOG parameter is forced in the DEFINE CLUSTER definition; if NO is specified, the CATALOG parameter is removed from the DEFINE CLUSTER definition.

Default: None

CISIZE=

Purpose: Specifies the value of the data or cluster and/or index control interval size in the corresponding component of the IDCAMS control card(s). Any existing specification of control interval size is overridden.

Syntax: CISIZE=*nnnnnn* or CISIZE=(*nnnnnn,nnnnnn*)

where the first *nnnnnn* is the data or cluster control interval size and the second *nnnnnn* is the index control interval size (0–999999).

Default: None

EVENTID=

Purpose: Specifies the identifier assigned to a user event in SMEVNTxx. This parameter will cause an event to be generated from this function.

Note: If EVENTID= is used on an FLST SET statement with MODE=INACT, the event will still be issued.

Syntax: EVENTID=xxxxxx

where xxxxx is the 5-character string specified on the EVNTID parameter in SMEVNTxx.

Default: None

ERASE=

Purpose: Specifies the removal of the ERASE parameter during cluster definition.

Syntax: ERASE=YES/NO

If YES is specified the ERASE parameter is forced in the DEFINE CLUSTER definition; if NO is specified, the ERASE parameter is removed from the DEFINE CLUSTER definition.

Default: None

IMBED=

Purpose: Specifies the removal of the IMBED parameter during cluster definition.

Syntax: IMBED=YES/NO

If YES is specified the IMBED parameter is forced in the DEFINE CLUSTER definition; if NO is specified, the IMBED parameter is removed from the DEFINE CLUSTER definition.

Default: None

OWNER=

Purpose: Specifies an owner to be assigned during cluster definition.

Syntax: OWNER=xxxxxxxxx...

where xxxxxxxxxx is a string of up to 8 characters or a MAINVIEW SRM keyword of up to 11 characters.

Default: None

PWDDEL=

Purpose: Specifies that any passwords specified for a VSAM definition be deleted.

Syntax: PWDDEL=YES/NO

The PWDDEL=YES is specified if all password specifications (CONTROLPW, MASTERPW, READPW, UPDATEPW) are removed from the cluster definition; if PWDDEL=NO is specified, any existing password specification is retained.

Default: PWDDEL=NO

REPL=

Purpose: Specifies the removal of the REPLICATE parameter during cluster definition.

Syntax: REPL=YES/NO

If YES is specified the REPLICATE parameter is forced in the DEFINE CLUSTER definition; if NO is specified, the REPLICATE parameter is removed from the DEFINE CLUSTER definition.

Default: None

REUSE=

Purpose: Specifies the removal of the REUSE parameter during cluster definition.

Syntax: REUSE=*YES/NO*

If YES is specified, the REUSE parameter is forced in the DEFINE CLUSTER definition; if NO is specified, the REUSE parameter is removed from the DEFINE CLUSTER definition.

Default: None

Parameters Not Supported

The following filter list parameters are not supported for VSAMCNTL:

- GDGVER
- VALUE

Usage Notes

The VSAM definition parameters ERASE, IMBED, and REPLICATE can be removed from a VSAM definition by specifying NO in the rule list. Alternatively, these VSAM parameters can be added to the VSAM file definition by specifying YES.

Note: The used space of the index component will increase in size when IMBED is removed.

The CATALOG=NO rule list parameter removes any CATALOG specification from the VSAM definition; if YES is specified, any existing catalog specification is retained.

The PWDDEL=YES rule list parameter directs that all VSAM definition passwords (CONTROLPW, MASTERPW, READPW, UPDATEPW) are removed from the VSAM definition. If NO is specified, any existing password specifications are retained.

The BUFSP and CISIZE rule list parameters allow the corresponding VSAM parameters to be overridden with the value specified on the MAINVIEW SRM rule list.

The OWNER parameter can be set to a 1–8 byte string that will be assigned during cluster definition. It can also be set to a 1–11 byte MAINVIEW SRM keyword whose value will be used as the owner identification.

Example

In this example

```
SET OWNER=USER
```

will direct MAINVIEW SRM to obtain the RACF user ID and use it as the owner ID.

```
SET OWNER=JOB
```

will direct MAINVIEW SRM to use the jobname as the owner ID.

If the length of value to be used for the owner identification exceeds 8 bytes, the remaining data is truncated.

The messages issued by VSAMCNTL are:

```
SVM3345I  job,step,dd,dsn REUSE SET TO yes/no
SVM3393I  job,step,dd,dsn CISIZE CHANGED FROM old TO new
SVM3395I  job,step,dd,dsn ERASE SET TO yes/no
SVM3396I  job,step,dd,dsn REPLICATE SET TO yes/no
SVM3397I  job,step,dd,dsn IMBED SET TO yes/no
SVM3394I  job,step,dd,dsn BUFSP SET TO nnnnn
SVM3398I  job,step,dd,dsn CATALOG SET TO yes/no
SVM3391I  job,step,dd,dsn PWDDEL SET TO yes/no
SVM3392I  job,step,dd,dsn OWNER SET TO xxxxx...
```

Example

This example shows you how to:

- Remove ERASE, REPLICATE, IMBED, and REUSE from all new VSAM cluster definitions and set the owner identification equal to the RACF user ID.
- Allow only systems programmers the ability to specify CATALOG or passwords.

SMFUNCA1 member

```
SET    NAME=VSAMCNTL ACTIVE=YES
        MSG=I SMF=I FLST=VC RLST=VC
        DESC='VSAM CONTROL'
```

The function is defined and active.

SMFLSTVC member

```
SET    MODE=ACT
        INC JOB=/'
```

Execute the function for all jobs.

SMRLSTVC member

```
SET    ERASE=NO REPL=NO IMBED=NO REUSE=NO X
        EXC OWNER=USER CATALOG=NO PWDDEL=YES
        INC USER=SYSP/
        JOB=/'
```

Disallow the use of ERASE, REPLICATE, IMBED, REUSE, CATALOG, and passwords for non-system programmers. Set the owner ID to the RACF user ID.

```
SET    ERASE=NO REPL=NO IMBED=NO REUSE=NO X
        INC OWNER=USER
        JOB=/'
```

Disallow the use of ERASE, REPLICATE, IMBED and REUSE for system programmers. Set the owner ID to the RACF user ID.

Chapter 4 How to Define Critical Data Set Lists

EasyPOOL allows you to identify a list of critical data sets in your environment that should never be placed on the same volume as any other data set in the list. It is important to consider, however, that CRITDSN is resource intensive and should be used only for a *small* list of critical data sets. It should not be used without considering the system impact.

To define critical lists, you must

- define a critical data set list (see “How to Use the CritList Option” on page 4-3 for instructions) and add it to the RLST
- define and activate the function(s) (DASDPOOL or SMSSELECT) with the critical data set list active
- activate the system member that includes the critical data set list information

You can change SMCRTxx parameters

- by editing the member directly
- by using the MAINVIEW SRM function SET statements
- through the **CritList** option on the Parmlib Members menu (see “How to Use the CritList Option” on page 4-3)

Example

This example shows you how critical data set list information is defined in each of the members. If you need further instructions, refer to the *MAINVIEW SRM User Guide and Reference*.

SMCRIT01 member		
SET	CRITDSN=PROD01 INC DSN=PROD.PARTS.DATABASE INC DSN=PROD.ASSEMBLE.DATABASE INC DSN=PROD.MASTER.DATABASE	Critical data set list PROD01 is created in SMCRIT01.
SMRLSTVC member		
SET	MODE=ACT VOLSEL=CRITDSN CRITBIAS=10 CRITFAIL=NO CRITLIST=PROD01 INC DSN=PROD.1	Critical data set list PROD01 is specified in the SMRLSTVC member along with other critical data set list processing parameters.
SMFLSTVC member		
SET	MODE=ACT INC JOB=PRODJOB1	Critical data set list PROD01 is specified in the SMRLSTVC member along with other critical data set list processing parameters.
SMFUNCAC member		
SET	NAME=DASDPOOL ACTIVE=YES MSG=I SMF=I FLST=VC RLST=VC DESC='DASD ALLOCATION'	The function is defined and active.
SMMSYS75 member		
SET	CRITLIST=01 FUNC=AC	The critical list suffix from SMCRIT01 is defined in SMMSYS75 as CRITLIST=01. the SMFUNCAC member is defined as FUNC=AC.

Note: You can also keep critical data sets from being allocated on the same EMC physical disk. See the DASDPOOL function and the CRITEMC parameter.

How to Use the CritList Option

Once you have identified the data sets that are critical at your data center, you may use the Critlist (EasyPOOL) option to define critical data set lists as instructed in the next table.

Step	Action
1.	Select Parmlib Members on the EZSRM Menu. A list of members displays.
2.	Select CritList from the selection list.
3.	Type E in the line command field of the CMD column next to an active SMCRTxx member or place your cursor on the member name, then press Enter .
4.	Edit the member to specify the data center critical data sets.
5.	Press PF3 to save and exit the member.
6.	Type R in the line command field of the CMD column next to the SMCRTxx member you edited, then press Enter . The member is refreshed. Note: By refreshing SMCRTxx, you have made it the active critical list. It will remain active until SVOS is stopped or until you refresh another critical list member.

Chapter 5 How to Define DFSMS Subpools

DFSMS-managed subpooling is a method by which DFSMS-managed Storage Groups, which are defined in ACS code and cannot be changed, can be subdivided into pools to improve data set allocation and performance. This functionality is limited to denying volumes outside of an DFSMS-defined pool. Volume selection cannot be directed outside of the data set's current Storage Group. If multiple pools are specified, the first pool with sufficient space for the data set is assigned. The first pool specified is the primary pool; all others are alternate pools.

To define DFSMS-managed subpools:

- Define the subpools within a Storage Group using the SMSPOLxx definition parameter member (see “How to Use the SMS Pools Option” on page 5-4).
- Activate the SMSSELCCT function in EasyPOOL using the SMFUNCxx definition parameter member (refer to the *MAINVIEW SRM User Guide and Reference* for instructions).
- Set system options SMS_ALLOC and SMS_EXTEND to Yes (refer to the *MAINVIEW SRM User Guide and Reference* for instructions).
- Assign SMSPOOLS to data sets with the parameters SMSPOOL and SMSPOOL_EXT (refer to the *MAINVIEW SRM User Guide and Reference* for instructions).

SMSPOLxx

SMSPOLxx is the DFSMS-managed subpool definition parameter member. It is read by MAINVIEW SRM during system start up. The suffix specification in SMMSYSxx indicates which version of SMSPOLxx contains SMS-managed subpool specifications for the particular configuration of MAINVIEW SRM that is being executed.

Parameters

Subpools are defined on SET statements. INC/EXC statements that follow a SET statement identify the devices to be included or excluded from a subpool.

You can change SMSPOLxx parameters

- by editing the member directly
- by using the MAINVIEW SRM system function SET statements
- through the **SMS Pools** option on the EZSRMP Parmlib Members menu (see “How to Use the SMS Pools Option” on page 5-4)

Parameter Quick Reference

SET statement pool parameters are described in the following table. Detailed descriptions of parameters can be found in the *MAINVIEW SRM User Guide and Reference*.

Table 5-1 SET Statement SMS-Managed Subpool Parameter Quick Reference

Parameter	Required	Description
POOLNAME=xxxxxxx	Yes	Name to be assigned to a pool

INC/EXC statement for SMS-managed subpool parameters are described in the following quick reference table.

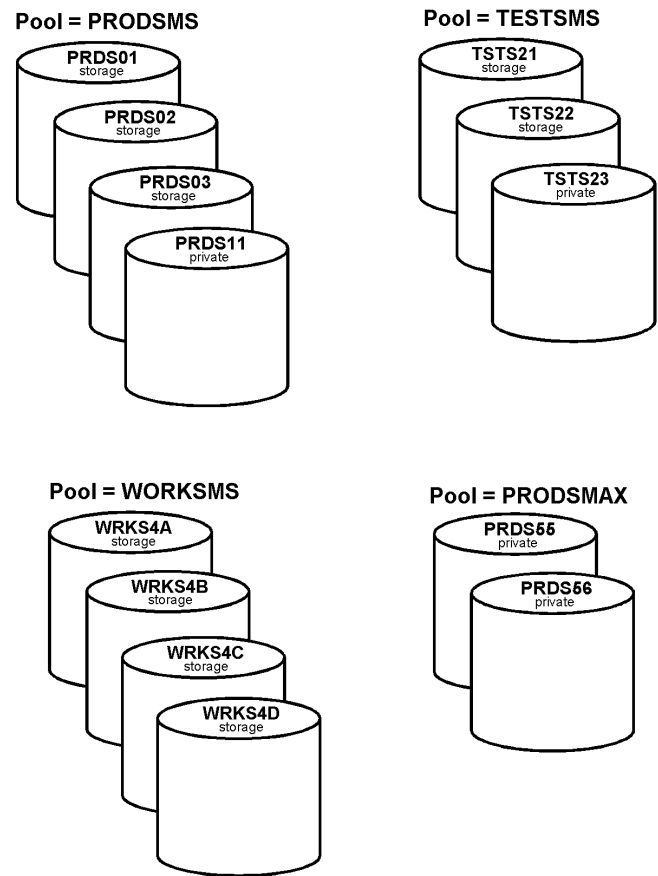
Table 5-2 INC/EXC Statement Pool Parameter Quick Reference

Parameter	Required	Description
VOL=xxxxxx	No	Volume serial number of device in pool
VOL=(xxxxxx,xxxxxx,...)	No	Multiple volume serial numbers (up to 15)

Example

This example shows you how to configure DFSMS-managed subpools using the information given in the following figure.

Figure 5-1 Possible System Storage Configuration for SMS-Managed Subpools



SMSPOL00 member	
SET POOLNAME=PRODSMS INC VOL=PRDS0/ INC VOL=PRDS11	The PRODSMS subpool is composed of all volumes whose names begin with PRDS0 and volume PRDS11.
SET POOLNAME=PRODSMAX INC VOL=(PRDS55,PRDS56)	The PRODSMAX subpool contains volumes PRDS55 and PRDS56.
SET POOLNAME=TESTSMS INC VOL=(TSTS21,TSTS22,TSTS23)	The TESTSMS subpool is composed of volumes TSTS21, TSTS22, and TSTS23.
SET POOLNAME=WORKSMS INC VOL=WRKS4/	The WORKSMS subpool contains all volumes whose names begin with WRKS4.

How to Use the SMS Pools Option

Once you have identified the criteria for SMS-managed subpools at your data center, you may use the SMS Pools option to define pools as instructed in the next table.

Step	Action
1.	Select Parmlib Members on the EZSRM Menu. A list of members displays.
2.	Select SMS Pools from the selection list.
3.	Type E in the line command field of the CMD column next to an active SMSPOLxx member or place your cursor on the member name, then press Enter .
4.	Edit the member to specify the data center pool definitions.
5.	Press PF3 to save and exit the member.
6.	Type R in the line command field of the CMD column next to the SMPOOLxx member you edited, then press Enter . The member is refreshed. Note: By refreshing SMSPOLxx, you have made it the active SMS-managed subpool list. It will remain active until SVOS is stopped or until you refresh another pool member.

Index

A

AC_CODE 3-59
ACF2USER 3-41
ACS routines
 testing 3-64
affinity separation for STK silos 3-112
ALCTYPE 3-104, 3-105
allocating data sets to DASD volume pools 3-3
ALLTAPE
 intercepting all tape generics 3-31, 3-44,
 3-58, 3-130
assigning tape data sets to pools 3-130
AVL 3-104, 3-105

B

BATCH 3-66
BESTFIT 3-20, 3-82
BLKINPUT 2-3
BLKOLDSR 2-3
BLKSIZE 3-54
blocksize 3-53
 optimizing for data sets 3-53
BUFSP 3-139, 3-140

C

CAL 3-59
CANDIDATE 3-5, 3-6
CATALOG 3-139, 3-140
changing allocation units to blocks 3-90

CHECK 2-3, 3-4
CISIZE 3-139, 3-140
 statement 3-140
clusters
 VSAM 3-18
COMP 3-125
CONTIG 3-104, 3-105
conversion to blocks 3-90
CRITBIAS 3-5, 3-6, 3-38, 3-39, 3-73, 3-74
CRITDSN 3-7, 3-20, 3-82, 3-83
CRITEMC 3-5, 3-7, 3-38, 3-39, 3-73, 3-75
CRITFAIL 3-5, 3-7, 3-38, 3-39, 3-73, 3-75
CRITLIST 2-3, 3-4, 3-5, 3-7, 3-38, 3-40, 3-73,
 3-75

D

DASD performance optimization 3-23, 3-84
DASDGENR 2-3, 3-4
DASDPOOL 3-3
data sets
 allocating with optimal blocksize 3-53
 allocation to virtual storage 3-136
 allocations, limiting space 3-95
 assign tape to pools 3-130
 DASD volume pooling 3-3
 data compaction for tape cartridge 3-125
 deletion, MODDELET 3-47
 empty, initializing 3-49
 expiration date 3-58
 forced cataloging 3-44
 limiting size 3-95, 3-104
 name checking operation 3-34

- permanent 3-95
- release space at close 3-100
- rename requiring management class change 3-68
- size 3-95, 3-104
- temporary 1-1, 3-34, 3-104
 - VIO allocation of 3-136
- utilities 3-101
- DATACLAS 3-66
- DD 3-66
- ddname 3-66
- defer tape mounts 3-128
- DEFUNIT 3-112, 3-113
- delete function
 - MODDELETE 3-47
- device 3-66, 3-90, 3-104, 3-125, 3-128, 3-130
- DFSMS enhancements
 - DSNCHECK, data set name checking 3-30
 - OPENEMPT, initialize empty data sets 3-49
 - OPTBLKSZ, allocate with optimal data set blocksize 3-53
 - SMSACSTE, test ACS routines 3-64
 - TAPEPOOL, non-DASD volume pooling 3-130
- DIR 3-104, 3-106
- DMYUNIT 2-3, 3-4
- DP_RENAME 2-3, 3-4
- DPO 3-20, 3-23, 3-82, 3-84
 - performance data display 3-25, 3-86
 - resources 3-24, 3-85
 - starting and stopping 3-23, 3-84
 - usage 3-24, 3-85
- DPORDEF 3-5, 3-8, 3-73, 3-76
- DPORMAX 3-5, 3-8, 3-24, 3-73, 3-76, 3-85
- DPORMIN 3-5, 3-8, 3-24, 3-73, 3-77, 3-85
- DPORSEP 3-5, 3-9, 3-24, 3-73, 3-77, 3-85
- DPOWIND 3-5, 3-9, 3-24, 3-74, 3-78, 3-85
- DSN 3-66
- DSNCHECK 3-30
 - operational method 3-34
- DSNn 3-31, 3-66
- DSNTYPE 3-66
- DSORG 3-66
- DSTYPE 3-66

E

- EasyPOOL functions
 - SETEXPDT 3-58
- ENVIR 3-66, 3-70
- ERASE 3-139, 3-141
- esoteric name 3-133
- EVENTID 3-5, 3-31, 3-38, 3-40, 3-44, 3-45, 3-47, 3-49, 3-54, 3-59, 3-60, 3-64, 3-68, 3-74, 3-78, 3-91, 3-96, 3-100, 3-104, 3-106, 3-112, 3-113, 3-118, 3-121, 3-125, 3-126, 3-128, 3-130, 3-131, 3-136, 3-139, 3-141
- EXPDT 3-59, 3-60, 3-66
- expiration date for data set 3-58

F

- FDRAISIT 3-38
- FORCE 3-54, 3-55
- FORCECAT 3-44
- forced cataloging
 - data sets 3-44
- forcing catalog disposition for new data sets 3-44
- functions
 - DFSMS enhancements
 - DSNCHECK 3-30
 - OPENEMPT 3-49
 - OPTBLKSZ 3-53
 - SMSACSTE 3-64
 - TAPEPOOL 3-130
 - space management
 - OPENEMPT 3-49
 - OPTBLKSZ 3-53
 - SPACCONV 3-90
 - SPACLIMI 3-95
 - SPACSQTY 3-104
 - TAPEDEFR 3-128
 - VIOALLOC 3-136
- standards control
 - DASDPOOL 3-3
 - FORCECAT 3-44
 - SUPJSCAT 3-118

G

generic name 3-66, 3-133

H

HDPODAYS 3-5, 3-12, 3-26, 3-74, 3-78, 3-87
HDPOETIM 3-5, 3-11, 3-26, 3-74, 3-79, 3-87
HDPORDEF 3-5, 3-9, 3-32, 3-74, 3-79
HDPORMAX 3-5, 3-10, 3-26, 3-74, 3-79, 3-87
HDPORMIN 3-5, 3-10, 3-26, 3-74, 3-80, 3-87
HDPORSEP 3-5, 3-11, 3-26, 3-74, 3-80, 3-87
HDPPOSTIM 3-5, 3-11, 3-26, 3-74, 3-87
HISTDAYS 2-3, 3-4, 3-26, 3-73, 3-87
HISTDPO 3-20, 3-25, 3-26, 3-82, 3-86, 3-87
historical DASD performance optimization 3-25
HLQ 3-66

I

IDRC 3-125, 3-130
IMBED 3-139, 3-141
initializing empty data sets 3-49
ISAM 3-55
ISPF interface
 performance data by pool 3-23, 3-84
ISPF service 3.2 3-101

J

JCLEXT 2-3, 3-4
JCLUREQ 2-3, 3-4
JOB 3-66
JOBACCTn 3-66
JOBCAT 3-118
JOBCAT DD
 suppressing 3-118

L

LIMIT 3-96
limiting space requested by users 3-95
LLQ 3-31, 3-32, 3-66

M

management class
 data set rename 3-68
MAXQLF 3-31, 3-32
MAXSIZE 3-17, 3-66, 3-95
MAXSPACE 3-20, 3-82
MGMTCLAS 3-66, 3-70
MINQLF 3-31, 3-33
MODDELET 3-47

N

non-DASD volume pooling 3-130
NQUAL 3-66
NVOL 3-5, 3-12, 3-20, 3-130
NVOLINDX 3-5, 3-12, 3-20
NVOLMAX 3-5, 3-12, 3-20

O

OPENEMPT 3-49
opening empty data sets 3-49
OPTBLKSZ 3-53
optimum blocksize 3-53
OWNER 3-139, 3-142

P

PERCENT 3-20, 3-82
permanent data sets 3-95
PGM 3-66
POOL 3-5, 3-13, 3-38, 3-40, 3-130, 3-131
pool
 non-DASD volumes 3-130
 volume pooling for DASD data sets 3-3
POOLNAME 5-2
 description 5-2
PQTY 3-104, 3-106
PROCOLD 2-3, 3-5
PWDDEL 3-139, 3-142

R

RACF 3-66
RAIDDEVTYPE 3-6, 3-13, 3-74, 3-80
RECORD 3-66
REJECT 2-3, 3-5, 3-6, 3-14, 3-19, 3-130, 3-132
releasing data set space at close 3-100
REPL 3-139, 3-142, 3-143
REPLACE 3-59, 3-60, 3-104, 3-107
response time 3-23, 3-84
RETPD 3-59, 3-61, 3-66
REUSE 3-139
RLSE 3-100, 3-101
ROUND 3-91, 3-104, 3-107

S

selecting volume from storage group 3-72
SEP 3-6, 3-14
SETEXPDT 3-58
setting data compaction for tape cartridge data set 3-125
setting expiration date for data set 3-58
setting optimum blocksize 3-53
setting primary and secondary space for data set 3-104
SIZE 3-66
SMPOOLxx subparameters
 POOLNAME 5-2
SMS_ALLOC 2-3, 3-73
SMS_EXTEND 2-3, 3-73
SMSACSTE 3-64
SMSMANAGED 3-64, 3-66
SMSMCREN 3-68
SMSPOOL 3-74, 3-81
SMSPOOL_EXT 3-74, 3-81
SMSSELECT 3-72
SPACCONV 3-72, 3-90
space allocation requests 3-95
space management
 OPENEMPT, initialize empty data sets 3-49
 OPTBLKSZ, allocate with optimal blocksize 3-53
 SPACLIMI, limits space allocation requests 3-95
 SPACSQTY, set primary and secondary space for data sets 3-104

TAPEDEFR, assign defer to tape allocations 3-128
VIOALLAC, control VIO allocation 3-136
SPACLIMI 3-95
SPACRLSE 3-100
SPACSQTY 3-104
SPACVOLA 3-6, 3-19, 3-121
SPLIT 3-112, 3-113
SQTY 3-104, 3-107
standards control
 DASDPOOL, volume pooling for DASD data sets 3-3
 FORCECAT, forced cataloging 3-44
 SUPJSCAT, suppressing JOBCAT, SUBCAT DDs 3-118
 TAPEPOOL, non-DASD volume pooling 3-130
STC 3-66
STEPACCTn 3-66
STEPCAT 3-118
STKSCR 2-3
STKSUPP 3-112
STOGROUP 3-66
STORCLAS 3-66
SUPJSCAT 3-118
suppressing JOBCAT and STEPCAT DDs 3-118
suppressing volume references 3-121
SUPVOL 3-6, 3-14, 3-22
SUPVOLRF 3-121

T

TAPECOMP 3-125
TAPEDEFR 3-128
TAPEGENR 2-3
TAPEPOOL 3-130
temporary data sets 1-1, 3-34, 3-104, 3-136
test ACS routines 3-64
trace 3-65
TRKCYL 2-3, 3-91, 3-92
TRKLEN 2-3, 3-91, 3-92
TSO 3-46, 3-66, 3-98

U

UNIT 3-6, 3-14, 3-17, 3-20, 3-66, 3-131, 3-132

USELIMIT 3-21
USER 3-66
USEVOL 3-6, 3-15, 3-19, 3-38, 3-41, 3-131,
3-132

V

VCOMPLLQ 3-31
 defined 3-33
VFORCE 3-6, 3-15, 3-20
VIO 3-136, 3-137
 allocating of temporary data sets 3-136
 controlling allocation 3-136
VIOALLOC 3-136
VOL 3-66, 5-2
VOLSEL 3-6, 3-16, 3-20, 3-26, 3-38, 3-41,
3-74, 3-82, 3-87
volume pooling
 non-DASD 3-130
volume selection 3-6, 3-16, 3-20, 3-41, 3-53,
3-54, 3-82, 3-90
VSAM 3-5, 3-18, 3-34, 3-50, 3-55, 3-61, 3-92,
3-97
VSAMCNTL 3-139

X

XMODE 3-66

END USER LICENSE AGREEMENT NOTICE

BY OPENING THE PACKAGE, INSTALLING, PRESSING "AGREE" OR "YES" OR USING THE PRODUCT, THE ENTITY OR INDIVIDUAL ENTERING INTO THIS AGREEMENT AGREES TO BE BOUND BY THE FOLLOWING TERMS. IF YOU DO NOT AGREE WITH ANY OF THESE TERMS, DO NOT INSTALL OR USE THE PRODUCT, PROMPTLY RETURN THE PRODUCT TO BMC OR YOUR BMC RESELLER, AND IF YOU ACQUIRED THE LICENSE WITHIN 30 DAYS OF THE DATE OF YOUR ORDER CONTACT BMC OR YOUR BMC RESELLER FOR A REFUND OF LICENSE FEES PAID. IF YOU REJECT THIS AGREEMENT, YOU WILL NOT ACQUIRE ANY LICENSE TO USE THE PRODUCT.

This Agreement ("**Agreement**") is between the entity or individual entering into this Agreement ("You") and BMC Software Distribution, Inc., a Delaware corporation located at 2101 CityWest Blvd., Houston, Texas, 77042, USA or its affiliated local licensing entity ("BMC"). "You" includes you and your Affiliates. "Affiliate" is defined as an entity which controls, is controlled by or shares common control with a party. THIS AGREEMENT WILL APPLY TO THE PRODUCT, UNLESS (1) YOU AGREED TO A WEB BASED LICENSE AGREEMENT WITH BMC WHEN ORDERING THE PRODUCT, IN WHICH CASE THAT WEB BASED LICENSE AGREEMENT GOVERNS THE USE OF THE PRODUCT, OR (2) IF YOU DID NOT AGREE TO A WEB BASED LICENSE AGREEMENT WITH BMC WHEN ORDERING THE PRODUCT AND YOU HAVE A WRITTEN LICENSE AGREEMENT WITH BMC, THEN THAT WRITTEN AGREEMENT GOVERNS THE USE OF THE PRODUCT. THE ELECTRONIC AGREEMENT PROVIDED WITH THE PRODUCT AS PART OF THE INSTALLATION OF THE PRODUCT WILL NOT APPLY. In addition to the restrictions imposed under this Agreement, any other usage restrictions contained in the Product installation instructions or release notes shall apply to Your use of the Product.

PRODUCT AND CAPACITY. "**Software**" means the object code version of the computer programs provided, via delivery or electronic transmission, to You. Software includes computer files, enhancements, maintenance modifications, upgrades, updates, bug fixes, and error corrections.

"**Documentation**" means all written or graphical material provided by BMC in any medium, including any technical specifications, relating to the functionality or operation of the Software.

"**Product**" means the Software and Documentation.

"**License Capacity**" means the licensed capacity for the Software with the pricing and other license defining terms, including capacity restrictions, such as tier limit, total allowed users, gigabyte limit, quantity of Software, and/or other capacity limitations regarding the Software. For licenses based on the power of a computer, You agree to use BMC's current computer classification scheme, which is available at <http://www.bmc.com> or can be provided to You upon request.

ACCEPTANCE. The Product is deemed accepted by You, on the date that You received the Product from BMC.

LICENSE. Subject to the terms of this Agreement, as well as Your payment of applicable fees, BMC grants You a non-exclusive, non-transferable, perpetual (unless a term license is provided on an order) license for each copy of the Software, up to the License Capacity, to do the following:

- (a) install the Software on Your owned or leased hardware located at a facility owned or controlled by You in the country where You acquired the license;
- (b) operate the Software solely for processing Your own data in Your business operations; and
- (c) make one copy of the Software for backup and archival purposes only (collectively a "**License**").

If the Software is designed by BMC to permit you to modify such Software, then you agree to only use such modifications or new software programs for Your internal purposes or otherwise consistent with the License. BMC grants You a license to use the Documentation solely for Your internal use in Your operations.

LICENSE UPGRADES. You may expand the scope of the License Capacity only pursuant to a separate agreement with BMC for such expanded usage and Your payment of applicable fees. There is no additional warranty period or free support period for license upgrades.

RESTRICTIONS: You agree to **NOT**:

- (a) disassemble, reverse engineer, decompile or otherwise attempt to derive any Software from executable code;
- (b) distribute or provide the Software to any third party (including without limitation, use in a service bureau, outsourcing environment, or processing the data of third parties, or for rental, lease, or sublicense); or
- (c) provide a third party with the results of any functional evaluation or benchmarking or performance tests, without BMC's prior written approval, unless prohibited by local law.

TRIAL LICENSE. If, as part of the ordering process, the Product is provided on a trial basis, then these terms apply: (i) this license consists solely of a non-exclusive, non-transferable evaluation license to operate the Software for the period of time specified from BMC or, if not specified, a 30 day time period ("**Trial Period**") only for evaluating whether You desire to acquire a capacity-based license to the Product for a fee; and (ii) Your use of the Product is on an AS IS basis without any warranty, and **BMC, ITS AFFILIATES AND RESELLERS, AND LICENSORS DISCLAIM ANY AND ALL WARRANTIES (INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT) AND HAVE NO LIABILITY WHATSOEVER RESULTING FROM THE USE OF THIS PRODUCT UNDER THIS TRIAL LICENSE ("Trial License").** BMC may terminate for its convenience a Trial License upon notice to You. When the Trial Period ends, Your right to use this Product automatically expires. If You want to continue Your use of the Product beyond the Trial Period, contact BMC to acquire a capacity-based license to the Product for a fee.

TERMINATION. This Agreement shall immediately terminate if You breach any of its terms. Upon termination, for any reason, You must uninstall the Software, and either certify the destruction of the Product or return it to BMC.

OWNERSHIP OF THE PRODUCT. BMC or its Affiliates or licensors retain all right, title and interest to and in the BMC Product and all intellectual property, informational, industrial property and proprietary rights therein. BMC neither grants nor otherwise transfers any rights of ownership in the BMC Product to You. BMC Products are protected by applicable copyright, trade secret, and industrial and intellectual property laws. BMC reserves any rights not expressly granted to You herein.

CONFIDENTIAL AND PROPRIETARY INFORMATION. The BMC Products are and contain valuable confidential information of BMC ("Confidential Information"). Confidential Information means non-public technical and non-technical information relating to the BMC Products and Support, including, without limitation, trade secret and proprietary information, and the structure and organization of the Software. You may not disclose the Confidential Information to third parties. You agree to use all reasonable efforts to prevent the unauthorized use, copying, publication or dissemination of the Product.

WARRANTY. Except for a Trial License, BMC warrants that the Software will perform in substantial accordance with the Documentation for a period of one year from the date of the order. This warranty shall not apply to any problems caused by software or hardware not supplied by BMC or to any misuse of the Software.

EXCLUSIVE REMEDY. BMC's entire liability, and Your exclusive remedy, for any defect in the Software during the warranty period or breach of the warranty above shall be limited to the following: BMC shall use reasonable efforts to remedy defects covered by the warranty or replace the defective Software within a reasonable period of time, or if BMC cannot remedy or replace such defective copy of the Software, then BMC shall refund the amount paid by You for the License for that Software. BMC's obligations in this section are conditioned upon Your providing BMC prompt access to the affected Software and full cooperation in resolving the claim.

DISCLAIMER. EXCEPT FOR THE EXPRESS WARRANTIES ABOVE, THE PRODUCT IS PROVIDED "AS IS." BMC, ITS AFFILIATES AND LICENSORS SPECIFICALLY DISCLAIM ALL OTHER WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT. BMC DOES NOT WARRANT THAT THE OPERATION OF THE SOFTWARE WILL BE UNINTERRUPTED OR ERROR FREE, OR THAT ALL DEFECTS CAN BE CORRECTED.

DISCLAIMER OF DAMAGES. IN NO EVENT IS BMC, ITS AFFILIATES OR LICENSORS LIABLE FOR ANY SPECIAL, INDIRECT, INCIDENTAL, PUNITIVE OR CONSEQUENTIAL DAMAGES RELATING TO OR ARISING OUT OF THIS AGREEMENT, SUPPORT, AND/OR THE PRODUCT (INCLUDING, WITHOUT LIMITATION, LOST PROFITS, LOST COMPUTER USAGE TIME, AND DAMAGE OR LOSS OF USE OF DATA), EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES, AND IRRESPECTIVE OF ANY NEGLIGENCE OF BMC OR WHETHER SUCH DAMAGES RESULT FROM A CLAIM ARISING UNDER TORT OR CONTRACT LAW.

LIMITS ON LIABILITY. BMC'S AGGREGATE LIABILITY FOR DAMAGES IS LIMITED TO THE AMOUNT PAID BY YOU FOR THE LICENSE TO THE PRODUCT.

SUPPORT. If Your order includes support for the Software, then BMC agrees to provide support (24 hours a day/7 days a week) ("Support"). You will be automatically re-enrolled in Support on an annual basis unless BMC receives notice of termination from You as provided below. There is a free support period during the one year warranty period.

(a) **Support Terms.** BMC agrees to make commercially reasonable efforts to provide the following Support: (i) For malfunctions of supported versions of the Software, BMC provides bug fixes, patches or workarounds in order to cause that copy of the Software to operate in substantial conformity with its then-current operating specifications; and (ii) BMC provides new releases or versions, so long as such new releases or versions are furnished by BMC to all other enrolled Support customers without additional charge. BMC may refuse to provide Support for any versions or releases of the Software other than the most recent version or release of such Software made available by BMC. Either party may terminate Your enrollment in Support upon providing notice to the other at least 30 days prior to the next applicable Support anniversary date. If You re-enroll in Support, BMC may charge You a reinstatement fee of 1.5 times what You would have paid if You were enrolled in Support during that time period.

(b) **Fees.** The annual fee for Support is 20% of the Software's list price less the applicable discount or a flat capacity based annual fee. BMC may change its prices for the Software and/or Support upon at least 30 days notice prior to Your support anniversary date.

VERIFICATION. If requested by BMC, You agree to deliver to BMC periodic written reports, whether generated manually or electronically, detailing Your use of the Software in accordance with this Agreement, including, without limitation, the License Capacity. BMC may, at its expense, audit Your use of the Software to confirm Your compliance with the Agreement. If an audit reveals that You have underpaid fees, You agree to pay such underpaid fees. If the underpaid fees exceed 5% of the fees paid, then You agree to also pay BMC's reasonable costs of conducting the audit.

EXPORT CONTROLS. You agree not to import, export, re-export, or transfer, directly or indirectly, any part of the Product or any underlying information or technology except in full compliance with all United States, foreign and other applicable laws and regulations.

GOVERNING LAW. This Agreement is governed by the substantive laws in force, without regard to conflict of laws principles: (a) in the State of New York, if you acquired the License in the United States, Puerto Rico, or any country in Central or South America; (b) in the Province of Ontario, if you acquired the License in Canada (subsections (a) and (b) collectively referred to as the "**Americas Region**"); (c) in Singapore, if you acquired the License in Japan, South Korea, Peoples Republic of China, Special Administrative Region of Hong Kong, Republic of China, Philippines, Indonesia, Malaysia, Singapore, India, Australia, New Zealand, or Thailand (collectively, "**Asia Pacific Region**"); or (d) in the Netherlands, if you acquired the License in any other country not described above. The United Nations Convention on Contracts for the International Sale of Goods is specifically disclaimed in its entirety.

ARBITRATION. ANY DISPUTE BETWEEN YOU AND BMC ARISING OUT OF THIS AGREEMENT OR THE BREACH OR ALLEGED BREACH, SHALL BE DETERMINED BY BINDING ARBITRATION CONDUCTED IN ENGLISH. IF THE DISPUTE IS INITIATED IN THE AMERICAS REGION, THE ARBITRATION SHALL BE HELD IN NEW YORK, U.S.A., UNDER THE CURRENT COMMERCIAL OR INTERNATIONAL, AS APPLICABLE, RULES OF THE AMERICAN ARBITRATION ASSOCIATION. IF THE DISPUTE IS INITIATED IN A COUNTRY IN THE ASIA PACIFIC REGION, THE ARBITRATION SHALL BE HELD IN SINGAPORE, SINGAPORE UNDER THE CURRENT UNCITRAL ARBITRATION RULES. IF THE DISPUTE IS INITIATED IN A COUNTRY OUTSIDE OF THE AMERICAS REGION OR ASIA PACIFIC REGION, THE ARBITRATION SHALL BE HELD IN AMSTERDAM, NETHERLANDS UNDER THE CURRENT UNCITRAL ARBITRATION RULES. THE COSTS OF THE ARBITRATION SHALL BE BORNE EQUALLY PENDING THE ARBITRATOR'S AWARD. THE AWARD RENDERED SHALL BE FINAL AND BINDING UPON THE PARTIES AND SHALL NOT BE SUBJECT TO APPEAL TO ANY COURT, AND MAY BE ENFORCED IN ANY COURT OF COMPETENT JURISDICTION. NOTHING IN THIS AGREEMENT SHALL BE DEEMED AS PREVENTING EITHER PARTY FROM SEEKING INJUNCTIVE RELIEF FROM ANY COURT HAVING JURISDICTION OVER THE PARTIES AND THE SUBJECT MATTER OF THE DISPUTE AS NECESSARY TO PROTECT EITHER PARTY'S CONFIDENTIAL INFORMATION, OWNERSHIP, OR ANY OTHER

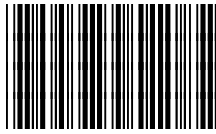
PROPRIETARY RIGHTS. ALL ARBITRATION PROCEEDINGS SHALL BE CONDUCTED IN CONFIDENCE, AND THE PARTY PREVAILING IN ARBITRATION SHALL BE ENTITLED TO RECOVER ITS REASONABLE ATTORNEYS' FEES AND NECESSARY COSTS INCURRED RELATED THERETO FROM THE OTHER PARTY.

U.S. GOVERNMENT RESTRICTED RIGHTS. The Software under this Agreement is "commercial computer software" as that term is described in 48 C.F.R. 252.227-7014(a)(1). If acquired by or on behalf of a civilian agency, the U.S. Government acquires this commercial computer software and/or commercial computer software documentation subject to the terms of this Agreement as specified in 48 C.F.R. 12.212 (Computer Software) and 12.211 (Technical Data) of the Federal Acquisition Regulations ("**FAR**") and its successors. If acquired by or on behalf of any agency within the Department of Defense ("**DOD**"), the U.S. Government acquires this commercial computer software and/or commercial computer software documentation subject to the terms of this Agreement as specified in 48 C.F.R. 227.7202 of the DOD FAR Supplement and its successors.

MISCELLANEOUS TERMS. You agree to pay BMC all amounts owed no later than 30 days from the date of the applicable invoice, unless otherwise provided on the order for the License to the Products. You will pay, or reimburse BMC, for taxes of any kind, including sales, use, duty, tariffs, customs, withholding, property, value-added (VAT), and other similar federal, state or local taxes (other than taxes based on BMC's net income) imposed in connection with the Product and/or the Support. This Agreement constitutes the entire agreement between You and BMC and supersedes any prior or contemporaneous negotiations or agreements, whether oral, written or displayed electronically, concerning the Product and related subject matter. No modification or waiver of any provision hereof will be effective unless made in a writing signed by both BMC and You. You may not assign or transfer this Agreement or a License to a third party without BMC's prior written consent. Should any provision of this Agreement be invalid or unenforceable, the remainder of the provisions will remain in effect. The parties have agreed that this Agreement and the documents related thereto be drawn up in the English language. Les parties exigent que la présente convention ainsi que les documents qui s'y rattachent soient rédigés en anglais.

SW EULA Int 030102

Notes



27843